REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
22-02-2011	Technical Paper	

4. TITLE AND SUBTITLE

Mission Capability Gains from Multi-Mode Propulsion Thrust Variations on a Variety Spacecraft Orbital Maneuvers

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Tiffany E Rexius (AFRL/RZST)

5d. PROJECT NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory (AFMC)
AFRL/RZST
4 Draco Drive
Edwards AFB CA 93524-7160

8. PERFORMING ORGANIZATION REPORT NUMBER

AFRL-RZ-ED-TP-2011-040

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory (AFMC)
AFRL/RZS
5 Pollux Drive
Edwards AFB CA 93524-7048

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S NUMBER(S)

AFRL-RZ-ED-TP-2011-040

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution A: Approved for public release; distribution unlimited (PA #10782).

13. SUPPLEMENTARY NOTES

Master's Thesis

14. ABSTRACT

Many spacecraft today have two separate propulsion systems: a chemical system for large maneuvers and an electric propulsion system for minor adjustments. The concept of coupling an electric and chemical propulsion system is not necessarily new but few studies have been published which discuss having an integrated chemical and electrical propulsion system. The utilization of a multi-mode propulsion (MMP) system could reduce the amount of propellant required for maneuvers in addition to lowering overall propulsion system mass. MMP refers to a propulsion system that couples an electric and chemical propulsion system which utilizes the same propellant tank and piping. A fully coupled system provides many advantages over a spacecraft with a single chemical or electric propulsion system. To help quantify the potential benefits in the utilization of a MMP system three separate spacecraft missions were analyzed. The missions were chosen to be representative of potential spacecraft missions. These missions include an altitude change mission, a phase change mission and a plane change mission. Each mission had a fixed propellant mass and time constraint. A variety of thrust profiles were investigated that utilized a combination of electric and chemical propulsion to complete the maneuver within the mission constraints. The "optimal" thrust profile for each mission was determined as the profile which completed the mission in the time constraint with the least amount of propellant.
In all cases the utilization of the electric propulsion system decreased overall required propellant mass.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Mike Holmes
Unclassified	**Unclassified**	**Unclassified**	SAR	161	**19b. TELEPHONE NUMBER** *(include area code)* N/A

Mission Capability Gains from Multi-Mode Propulsion Thrust Variations on a Variety

Spacecraft Orbital Maneuvers

An Independent Study Report

Presented to the

Faculty of

California State Polytechnic University, Pomona

In Partial Fulfillment

Of the Requirements for the Degree

Masters of Science

In

Engineering

By

Tiffany Elizabeth Rexius

March 2011

Signature Page

INDEPENDENT STUDY: Mission Capability Gains with Multi-Mode Propulsion Thrust Variations on a Variety of Spacecraft Orbital Maneuvers

AUTHOR: Tiffany Rexius

DATE SUBMITTED: ____

Dr. Michael Holmes

Ind. Study Committee Chair ____

Air Force Research Laboratory

Edwards AFB, CA

Dr. Ali Ahmadi ____

Aerospace Department

Contents

Abstract

Many spacecraft today have two separate propulsion systems: a chemical system for large maneuvers and an electric propulsion system for minor adjustments. The concept of coupling an electric and chemical propulsion system is not necessarily new but few studies have been published which discuss having an integrated chemical and electrical propulsion system. The utilization of a multi-mode propulsion (MMP) system could reduce the amount of propellant required for maneuvers in addition to lowering overall propulsion system mass. MMP refers to a propulsion system that couples an electric and chemical propulsion system which utilizes the same propellant tank and piping. A fully coupled system provides many advantages over a spacecraft with a single chemical or electric propulsion system.

To help quantify the potential benefits in the utilization of a MMP system three separate spacecraft missions were analyzed. The missions were chosen to be representative of potential spacecraft missions. These missions include an altitude change mission, a phase change mission and a plane change mission. Each mission had a fixed propellant mass and time constraint. A variety of thrust profiles were investigated that utilized a combination of electric and chemical propulsion to complete the maneuver within the mission constraints. The "optimal" thrust profile for each mission was determined as the profile which completed the mission in the time constraint with the least amount of propellant. In all cases the utilization of the electric propulsion system decreased overall required propellant mass.

List of Tables

List of Figures

List of Symbols

Symbol	Description	Units
e	Eccentricity	N/A
g	Acceleration due to Gravity	m/s^2
i	Inclination	rad
I_{sp}	Specific Impulse	sec
m_i	Initial Mass	kg
m_p	Propellant Mass	kg
m_{sc}	Spacecraft Mass	kg
n_{rot}	Number of Rotations	N/A
p	Semi-latus Rectum	km
P_{init}	Period of the Initial orbit	sec
P_{trans}	Period of the Transfer orbit	sec
T	Thrust	N
t_{burn}	Thrust Time	sec
Ω	Right Ascension of the Ascending Node (RAAN)	rad
ω	Rotation Rate	rad/sec
η	True Anomaly	rad
δ	Plane Change Angle	rad
μ	Gravitational Parameter	km^3/s^2
θ	Rephase Angle	rad

ΔV	Change in Velocity	m/s
$\vec{r}$	Position Vector	km
$\overrightarrow{r_{pqw}}$	Position Vector in Perifocal Reference Frame	km
$\overrightarrow{r_{\imath\jmath k}}$	Position Vector in Geodetic Equatorial Reference Frame	km
$\overrightarrow{g_{\imath\jmath k}}$	Gravity Vector in Geodetic Equatorial Reference Frame	km
$\vec{V}_1$	Initial Orbit Velocity Vector	km/s
$\vec{V}_2$	Transfer Orbit Velocity Vector	km/s
$\vec{V}_{diff}$	Difference between Final and Transfer Velocity Vector	km/s
V_{r1}	Radial Component of Initial Orbit Velocity	km/s
V_{r2}	Radial Component of Transfer Orbit Velocity	km/s
$V_{\perp 1}$	Perpendicular Component of Initial Orbit Velocity	km/s
$V_{\perp 2}$	Perpendicular Component of Transfer Orbit Velocity	km/s

Acknowledgments

I would like to thank my advisor, Dr. Mike Holmes, for his instruction and support as well as the teachers from the Cal Poly Pomona Program. I would also like to thank my husband for helping me stay focused and encouraging me throughout the program.

Introduction

Three separate spacecraft missions were analyzed utilizing a multi-mode propulsion system[1]. The missions chosen were selected to represent a broad range of common orbital maneuvers. The missions included a phase change mission, an altitude change mission and a plane change mission. An extensive literature search was performed that did not yield any papers that addressed the combined use of an electric and chemical propulsion system to perform spacecraft maneuvers.

Edelbaum mentioned the benefits from combining a low and high thrust system to perform orbital maneuvers in concluding remarks in a 1962 *Journal of Astronautical Sciences* article but was left to future work[6]. The lack of analysis for missions utilizing electric propulsion was largely due to the lack of computing power. Numerous iterations are required to analyze an electric propulsion transfer which would have been incredibly time consuming in the 1950's and 1960's when Edelbaum published his paper.

Advancements in computing power have made it possible to perform simulations that involve numerous iterations in a relatively short period of time. These advancements have made it possible to model low thrust orbital transfers. The orbit transfers modeled for this paper required on the order of 10,000+ iterations; these missions took several hours for MATLAB®* to compute.

The differences between a less accurate "impulsive" orbit updating method was compared with a method that updated the spacecraft's position and velocity continuously throughout the transfer orbit for the plane change mission.

*MATLAB is a registered trademark of The Mathworks, Inc., One Apple Hill, Natick, MA.

The only method described in any of the textbooks found for performing a plane change maneuver was an impulsive cranking maneuver performed at the crossover point of the initial orbit and the desired orbit[2-5].

The equations utilized in the separate mission analysis were based on equations from several textbooks including Fundamentals of Astrodynamics by Bate, Mueller, and White[2] as well as Orbital Mechanics for Engineering Students by Curtis[3]. Numerous other textbooks were referenced[4,5]. Additional equations were developed for the specific spacecraft missions analyzed. All the equations were programmed with MATLAB® to simulate the orbital transfer trajectories. The data was compiled and plots were generated with MATLAB® as well as Excel.

Overview

Spacecraft chemical propulsion systems provide high thrust resulting in fast trip time but at the cost of the propellant required. Spacecraft electric propulsion systems provide high specific impulse which result in low propellant usage but have long trip times, sometimes on the order of years. The combined use of chemical and electric propulsions system for spacecraft, though it adds weight and complexity, has been considered beneficial to spacecraft performance based on the propellant mass savings. Presently this involves having entirely separate propulsion systems. Many spacecraft utilize a chemical propulsion system for primary mission requirements and use a separate electric propulsion system for station-keeping maneuvers. This reflects the propellant

mass savings of electric propulsion over chemical propulsion systems. The savings resulting from electric propulsion must outweigh the added complexity and mass.

The idea of coupling an electric and chemical propulsion system through the use of a common propellant to decrease propulsion system mass and decrease complexity has only recently been investigated[1]. The concept has been mentioned but almost no results or studies have been published on development results. Overall propulsion system mass may be reduced by integrating the chemical and electric propulsion systems through the use of a common propellant. Chemical and electric propulsion systems utilizing the same propellant enables them to use the same propellant tank and common piping. Having one propellant tank and common piping decreases the overall propulsion system mass. This set up is referred to as a multi-mode propulsion (MMP) system. An example of a MMP system is shown in Figure 5 in the spacecraft specifications subsection.

Each of the three missions modeled in this paper had a time and propellant constraint that would be challenging to both the chemical and the electric propulsion systems. Missions with small time constraints were modeled primarily using chemical propulsion supplemented by electric propulsion to lower the required propellant. The time constraint was also modified to show how longer time constraints can decrease propellant mass when the electric propulsion system was able to thrust for additional periods of time.

To determine the mission tradespace three missions were analyzed which represent a variety of missions that a spacecraft may be required to perform. An orbit raise and lower mission, a phase change mission and a plane change mission were analyzed. The electric propulsion system was unable to achieve the trip time constraint

for the missions and the chemical system utilized the majority of the propellant mass available or was unable to achieve the maneuver within the propellant mass constraint. An "optimal" combination of electric and chemical propulsion system thrust was found for each mission that resulted in the time constraint being achieved with the lowest amount of propellant through the modeling of numerous thrust profiles.

As shown in the upcoming literature review, no papers were found which looked at a coupled chemical-electric propulsion system to complete maneuvers. If an electric propulsion system was analyzed it was considered completely separated from the chemical system. Electric propulsion systems must be analyzed in conjunction with chemical propulsion systems for its potential to be realized. Treating electric propulsion systems separately causes it to be disregarded for missions with tight time constraints when it could potentially decrease the propellant usage. The utilization of a multi-mode propulsion system causes the electric and chemical propulsion systems to be treated as an integrated unit. The shifting design of the propulsion system could lead mission analysts or engineers to begin using both modes of the multi-mode system to complete maneuvers.

Before work began on analyzing the missions and developing the necessary equations, a literature search was performed. There was little to no information available regarding multi-mode propulsion and hardly any information regarding the modeling of spacecraft maneuvers utilizing electric and chemical propulsion. The majority of papers that resulted from multi-mode propulsion searches had different definitions of "multi-mode". The most common definition was a combination of air-breathing and liquid or solid propulsion, such as the X-51. There were quite a few papers on developments in electric propulsion and on orbit testing of hardware. There were also no papers which

discussed combined high and low thrust propulsion systems completing orbital maneuvers. A few papers had general conclusions but with no explanation on how the conclusions were reached. The following section serves as a general overview on what is available regarding chemical and electric propulsion systems and mission analysis.

Literature Review

Electric thruster design and experimentation first occurred in the late 1940's. Since then numerous papers have been published which address electric propulsion system development and experimentation. Warner Von Braun expressed interest in electric propulsion and prompted a series of papers to be published beginning in 1954 regarding electric propulsion systems[6]. In 1962 Edelbaum published a paper in the *Journal of Astronautical Sciences* that broached the idea of coupling an electric and chemical propulsion system[7]. In the paper he mentioned how coupling the two propulsion systems could be very beneficial for on-orbit maneuvers but no analysis or results were discussed.

Electric Propulsion Development & History

The first operational flight test of an electric propulsion system was in 1964 conducted by Russia and used a Teflon-pulsed plasma system for attitude control[6,8]. Much development has occurred since then by numerous aerospace companies as well as governments[9-13]. Russia and the United States were the primary countries to invest time and resources into electric propulsion development. Stuhlinger wrote a thorough background regarding the origin of electric propulsion and its development[5].

The realization that electric propulsion could result in reduced fuel consumption has resulted in many spacecraft in orbit today with some form of an electric propulsion system[6]. These systems mainly perform orbit maintenance tasks such as station keeping. Electric propulsion has gained acceptance over the years in the spacecraft community by being utilized for station-keeping and small orbital adjustments. Its slow acceptance was due largely to the amount of power required for most electric propulsion systems, the weight associated with the components necessary to convert the power into a useful form for propulsion and unknown field effects. Improvements to spacecraft power systems are positively impacting electric propulsion systems[11].

There are many different variations of thrusters that are classified as electric propulsion thrusters. Three different types of electric propulsion thrusters were analyzed in a 1994 paper by Cassady et al.[9]. The thrusters were a 30 cm derated ion thruster, a xenon Hall thruster (SPT-100), and a hydrazine arcjet thruster. Repositioning a spacecraft in a sun synchronous low earth orbit and an east-west location change in GEO were addressed using both chemical and electric propulsion separately. The lowest electric propulsion system mass resulted from the hydrazine arcjet thruster because it utilized the same propellant as the chemical propulsion system. This type of a shared propellant system between the chemical and electric propulsion system is referred to as a multi-mode, or dual-mode, propulsion system. This was the only paper which mentioned a combined system, but the two modes were not used together for the mission analysis. The LEO and GEO missions were investigated using either the electric propulsion system or the chemical system alone. The authors reached several conclusions including that LEO orbital mechanics can be more complex than in GEO. Also, inclination change

maneuvers are extremely costly in regards to delta v as compared with an orbit raising or lowering maneuver. The authors did not model the electric propulsion system performing the inclination change because it was assumed that spacecraft could only thrust at the crossover nodes. This impulsive thrusting at the crossover node is the only transfer method documented in textbook equations for completing plane changes. Development of the commonly used equations are reviewed in Equations 23 – 30 of this paper. The equations provide an estimate for the required delta-v, but give no time estimates because the maneuver is assumed impulsive. Due to this electric propulsion systems are not modeled to perform plane change maneuvers.

Of the three separate electric propulsion systems the arcjet performed the maneuvers the quickest, followed by the hall thruster, then the ion thruster. The ion thruster was able to enable 11 maneuvers in GEO, whereas the hydrazine arcjet enabled 3 maneuvers. The trip time for each ion maneuver was much longer than the arcjet trip time; this was a direct result of the specific impulse and power available.

A paper written in 2006 by Pidgeon et al. reviewed the performance of the Stationary Plasma Thruster (SPT-100)[10]. The SPT-100 is a Hall Effect type thruster that was developed in Russia and was used on a western spacecraft for the first time 2004. This thruster is very popular for electric propulsion systems and is referenced many times in literature. The pre-flight expectations of the SPT-100 were compared with on-orbit performance. At the time of this report the SPT-100 thrusters had been firing twice a day for two years. During that duration the thruster performed without fault and has since been incorporated onto 19 other spacecraft as of 2006.

Martinez-Santchez et al. noted that hydrazine was the most successful and popular for testing with resistojets and arcjets, though numerous propellants were tested[8]. Three missions that electric propulsion has primarily been used for are North-South Station keeping (NSSK), orbit raising or lowering, and orbital repositioning[14]. NSSK for geosynchronous satellites can require a large delta-v allotment. Earth-moon effects can require around 51 m/s/year which can become a large consideration over a 15 year design life of a satellite. An entirely electric propulsion transfer from Low Earth Orbit (LEO) to Geostationary Orbit (GEO) can require over 1000 m/s additional delta v than a chemical system but require less propellant making it a viable system if trip time is not an issue.

Electric propulsion is being increasingly considered for larger orbit maneuvers which have previously been completed purely by chemical propulsion systems. These maneuvers can be accomplished with a substantially smaller amount of propellant than required by a chemical propulsion system to perform the same maneuver. A downside to having an electric propulsion system in addition to a chemical propulsion system is that it causes increased weight and complexity because the spacecraft would essentially have two separate propulsion systems on board. The benefits of electric propulsion outweigh the added complexity in most cases such that there have been a variety of papers written on the subject of electric propulsion and track its development throughout history[5,9].

A study performed in 2008 investigated 3 separate missions in which small satellites in LEO performed maneuvers using an electric propulsion system[18]. The three separate missions were an altitude change, an inclination change, and a phase change. A specific electric propulsion system was not studied; a range of specific impulse values from 1,000 – 3,000 sec was used. Of the three missions investigated electric propulsion

did not perform well for a large (> 90 deg) inclination change maneuver. One of the main drawbacks was that for many cases the transfer time would take over a year which may not be acceptable for certain missions. This paper did not perform a detailed analysis using electric propulsion, but calculated the expected delta-v to perform the maneuvers based on generic textbook equations then varied the specific impulse based on power available.

A 1993 paper suggested that an electric propulsion system could potentially create a new realm of "designer" orbits[15]. These orbits could be achieved through a continually thrusting electric propulsion system to maintain out of the ordinary orbits, such as molniya-like orbits, or satellites flying in formation. The author proposed that electric propulsion, besides performing regular operations, could be used to increase the overall capability and mission trade space available with chemical propulsion.

Multi-Mode Propulsion Development

Multi-mode propulsion has developed into two separate branches. One branch has thrusters such as Aerojet's BPT-4000, which has a high thrust/high power mode and a low thrust/low power mode. The maximum thrust on this system was around 0.3 N and has had extensive testing[11]. The testing of the BPT-4000 has confirmed a life capacity of over 7000 hours as of 2002 when the paper was published. The BPT-4000 was developed by Aerojet which has been developing hall thrusters since 1994 and have been integrated onto numerous spacecraft platforms including National Reconnaissance Office and NASA satellites.

A separate branch of multi-mode propulsion had an electric thruster and a separate chemical thruster that share an integrated propulsion system with a common propellant[7]. The propulsion system can switch from chemical to electric at any time causing the maximum thrust to be that of the chemical thruster. Hydrazine is the primary propellant that has been considered. This system has a decreased propulsion system weight due to the utilization of a common propellant tank and piping. Utilizing this set up can potentially extend the lifetime of the satellite as well as provide additional mission capability. Since multi-mode propulsion systems are relatively new to space propulsion there are few papers available regarding spacecraft multi-mode propulsion, but there have been many studies performed that have laid the groundwork.

Multi-Mode Propulsion Mission Analysis

Many studies have been performed which investigate the advantages of having a separate chemical and electric propulsion system on a spacecraft to provide the fast transfer time with the fuel economy of an electric propulsion system. The first published study that mentioned a combined chemical and electric propulsion system for performing maneuvers was by Edelbaum in 1962[7]. Up until that point chemical and electric systems had always been discussed as two completely separate systems. The chemical system would perform large maneuvers to achieve the desired orbit and electric propulsion would be used for orbit maintenance. Edelbaum concluded that the use of chemical and electric propulsion together for a single maneuver could provide an increase in payload but no study results were presented. This was largely due to the extensive computation required to model an electric propulsion transfer of any extended duration.

A study presented at the AIAA International Communications Satellite Systems Conference in 1996 evaluated the use of chemical and electric propulsion systems for the repositioning of Geostationary spacecraft[16]. A generic hydrazine chemical propulsion system was analyzed alongside several electric propulsion systems including a hydrazine arcjet, a stationary plasma thruster (SPT-100), and a xenon ion propulsion system (XIPS-25). Relocating the spacecraft was analyzed separately for the chemical system and then for the electric system with no combined maneuvers. Electric propulsion systems build up delta-v gradually due the extended thrusting times and is a known penalty. The largest trade off between electric propulsion and chemical propulsion for maneuvers was decreased trip time or decreased propellant mass.

A study documented in the *Journal of Spacecraft and Rockets* in 2002 investigated the advantages of having a chemical and electric propulsion system for a LEO to GEO transfer[19]. Two separate trades were performed to look at optimizing the trip time as well as fuel consumption. The chemical system analyzed was similar to the space shuttle orbital maneuvering system and used N2O4/MMH. The electric propulsion thrusters analyzed were 30-kW arcjet thrusters based on the electric propulsion space experiment (ESEX) which used ammonia, a stationary plasma thruster (SPT-100), and a Hall Effect Thruster (HET-200) which both used xenon as a propellant. One issue that was addressed was that for LEO to GEO transfers using a pure electric propulsion system the total radiation dose becomes high due to the long trip duration. They suggested using chemical propulsion to decrease the trip time and exposure in the Van Allen belts. To decrease the radiation dosage an electric/chemical/electric thrust profile was the best for a LEO to GEO transfer.

Chemical and electric propulsion have been analyzed for many lunar missions. A study by Kluever in the *Journal of Spacecraft and Rockets* in 1994 described using a chemical propulsion system for transfer from a translunar injection to a lunar orbit insertion (LOI) with electric propulsion used from LOI to a final polar, circular low lunar orbit[17]. The final delta-v increase to escape the moon's gravity field is suggested as being done by the electric propulsion system. Having an electric propulsion system perform some of the orbital maneuvering tasks created a 15% higher payload capability for the spacecraft to complete its mission. An increase in mission payload of that amount is very substantial.

Conclusion

Multi-mode propulsion is a very new topic and only a few papers address the possibility[17,20,21]. The acceptance of multi-mode propulsion will become more prevalent as the advantages of a combined system are published. Currently the majority of papers which discuss a chemical/electric system do not use any combined thrust profiles for the missions analyzed. Modeling extended duration electric propulsion transfers has been made much more practical with improvements in computing power. Ultimately electric propulsion systems provide the ability to increase the life of a satellite due to more conservative fuel use in performing orbital maneuvers.

Mission Overview

The missions analyzed represent a wide variety of orbital maneuvers that a spacecraft may be required to perform in its lifetime. The missions could be performed

separately or in conjunction with one another. The depiction of each mission analyzed is shown in the following figures.

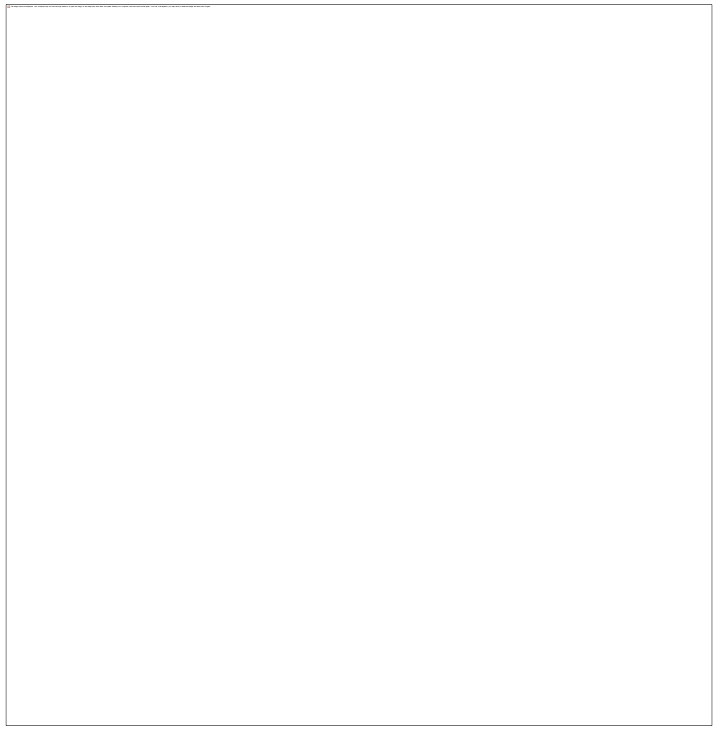

Figure 1: Mission 1 Depiction

The first mission was a 180 degree phase change mission. A phase change maneuver is where the spacecraft is required to be in the same orbit it currently is in, but be in a different location in that orbit. This requires a change in the true anomaly of the spacecraft. A phase change of 180 deg implies that the spacecraft is required to be on the other side of the orbit than it currently is. The spacecraft had to perform the phase change in 12 hours and return to its initial position in 30 days. This mission is shown in Figure 1.

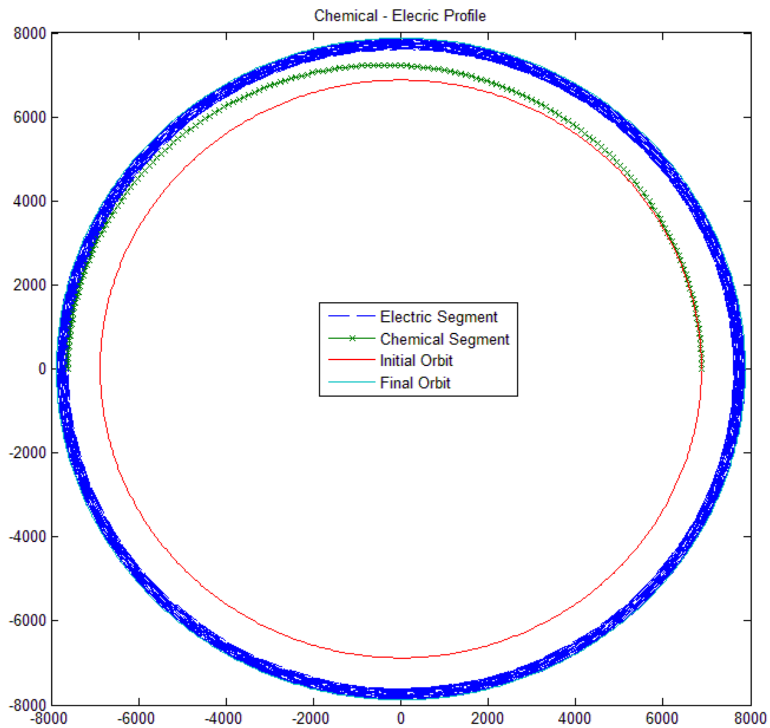

Figure 2: Mission 2 Depiction

The second mission involved a spacecraft increasing its orbit by 1000 km in 12 hours then returning to the initial orbit in 30 days. This mission is depicted in Figure 2. The thrust profile shown was for an initial chemical thrust portion followed by the electric propulsion system performing the remainder of the altitude change maneuver within the 12 hour time constraint.

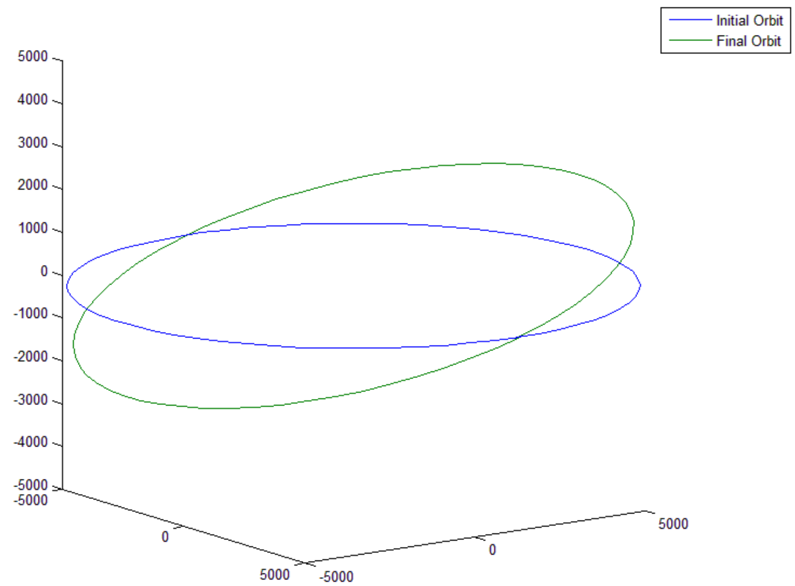

Figure 3: Mission 3 Depiction

The third mission was a 15 deg plane change mission. Electric propulsion systems are normally never modeled to perform a plane change maneuver. Plane change maneuvers are generally completed with an impulsive burn at the crossover location between the two orbits. This occurs twice an orbit at the periapsis and the apoapsis. The initial and final orbit are depicted in Figure 3. The impulsive method would cause the spacecraft to transfer "instantaneously" into the final orbit. The following figure depicts what a low thrust plane change looked like for Mission 3.

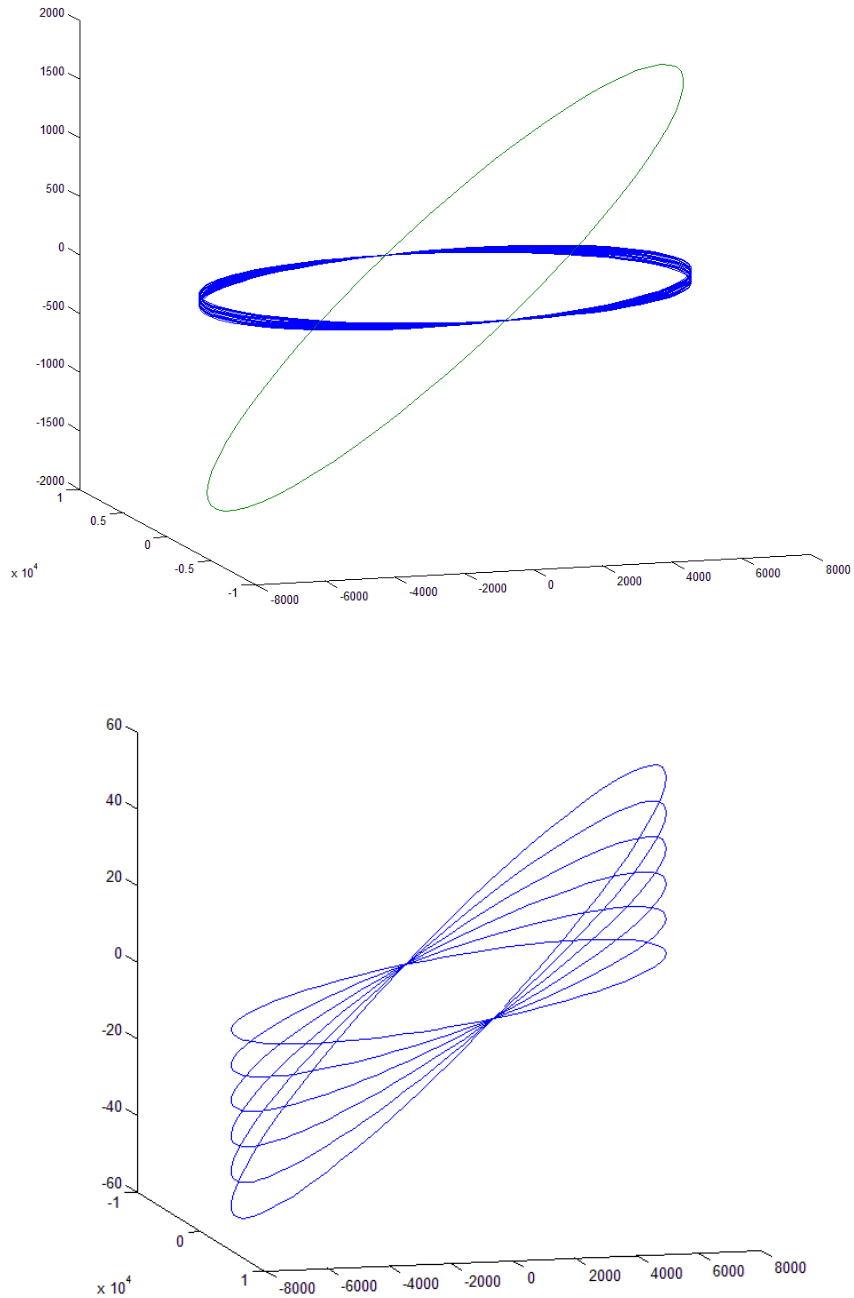

Figure 4: Mission 3 Low-Thrust Depiction

The electric propulsion would only be able to apply a small amount level of thrust; 0.11 N for the system analyzed in this paper. The amount of time the electric propulsion system was able to thrust for was varied, and the thrust was applied in a specific location

to leave the orbit parameters the same and change only the inclination. Equations were develop to analyze all the missions and were programmed using MATLAB. The codes written for each mission are included in the appendices.

Spacecraft Specifications

The spacecraft modeled had a total mass of 180 kg with 80 kg allotted for propellant. The mass of specific components were not calculated. The spacecraft had a multi-mode propulsion (MMP) system consisting of a chemical portion that had a specific impulse (Isp) of 235 sec which provided a thrust of 22 N. The electric propulsion system had an Isp of 600 sec and produced a thrust of 0.11 N. A schematic of this propulsion system is shown in Figure 5.

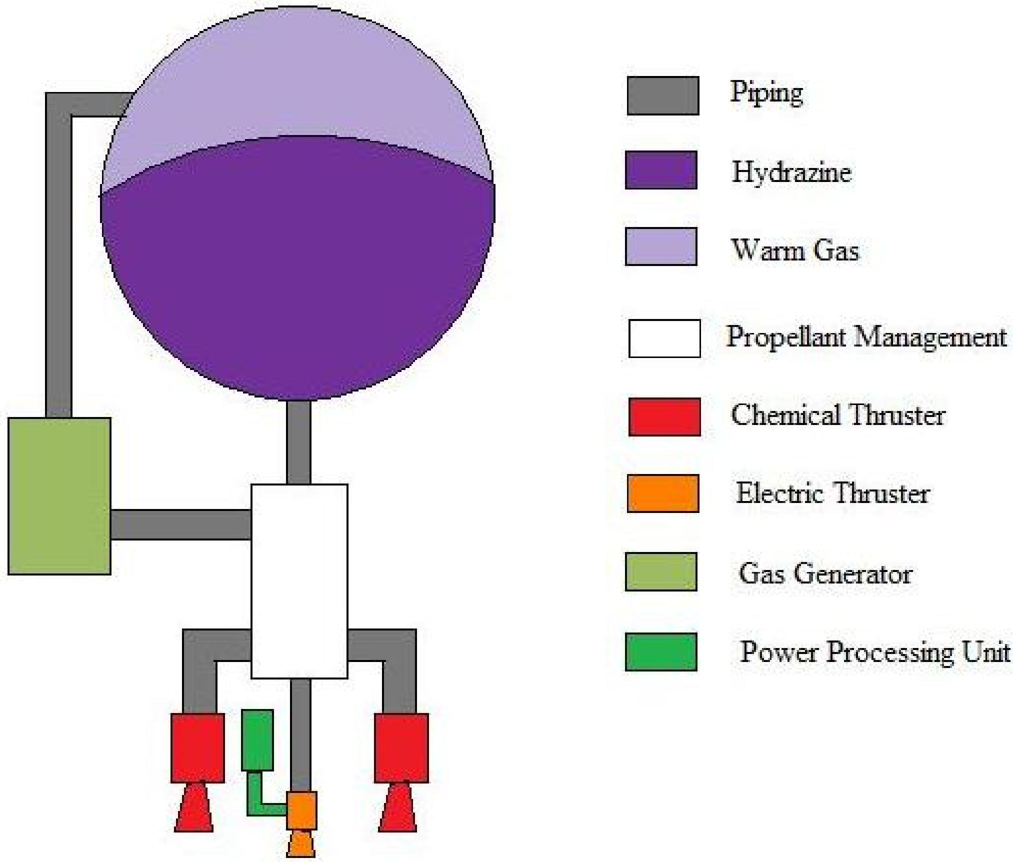

Figure 5: Multi-Mode Propulsion Set Up

As seen in Figure 5 the electric propulsion thrusters have a separate power processing unit but share a common propellant tank and piping up until the propellant management system. Combining the systems this way makes the most use of the common elements, such as the gas generator, required by both systems. Many different propellants could be used besides hydrazine, but the specific impulse and thrust values used the mission analysis were representative of a hydrazine system. To analyze a system with a different propellant the specific impulse and thrust values could easily be changed.

The detailed spacecraft propulsion system component weight and thrust based on power was not included in this analysis but is commented on in the Future Work section.

The required propellant mass and trip time for only the chemical system or only the electric system were compared with the combined system with various thrust profiles.

The 180 kg maximum spacecraft mass was chosen because it is the maximum weight allowed on an Evolved Expendable Launch Vehicle (EELV) Secondary Payload Adapter (ESPA) ring. The ESPA ring allows secondary payloads to utilize potentially unused space between the primary payload on the Evolved Expendable Launch Vehicle and on the upper stage. The ESPA ring has been utilized on the Atlas V, but could also be used on the Delta IV. Enforcing mass constraints imposed by the ESPA ring provides a more understandable basis for the capability modeled. Having a constrained mission space promotes the utilization of the most optimal orbit transfers for the spacecraft.

Initial Orbit Constraints

The spacecraft was initially located in a circular orbit at an altitude of 500 km. The initial inclination was zero degrees, which only affected one of the missions analyzed. The drag on the spacecraft was calculated as being minimal at 500 km and as such was not included in any necessary change in velocity (delta-v) allotment calculations. The other orbital parameters, right ascension of the ascending node (RAAN or Ω), eccentricity (ecc), and true anomaly (η) were all modeled initially as zero. Since the orbit modeled was circular the argument of perigee (ω) was undefined. The code was written such that the orbital elements could be updated depending on the orbit, but were not specified for the missions analyzed.

Reference Frame Usage

The Perifocal Reference Frame was used for missions 1 and 2. A diagram of the Perifocal Coordinate system, the Geocentric-Equatorial Reference frame and the reference frame depicted on one another is shown below. The following figures are from *Fundamentals of Astrodynamics* by Bate, Mueller, and White[2].

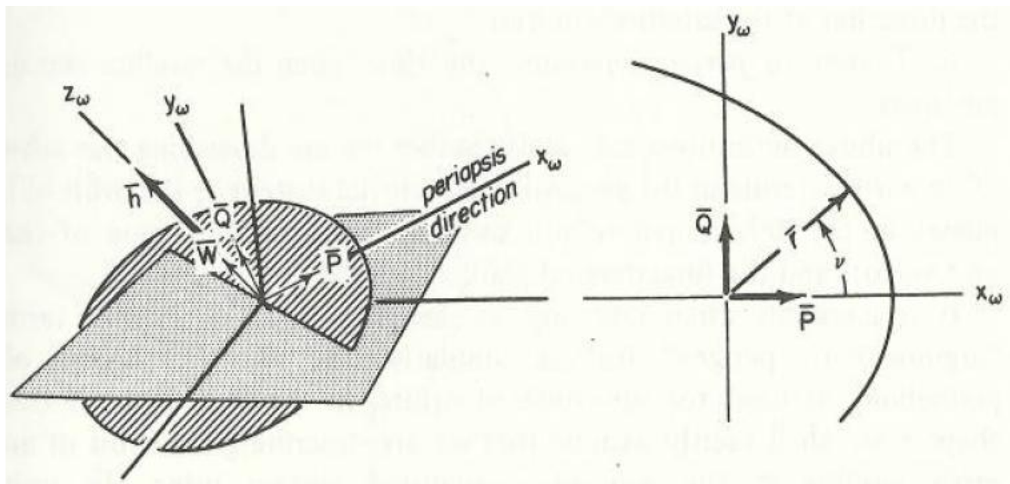

Figure 6: Perifocal Reference Frame[2]

This system was used because there were no out of plane maneuvers required for the altitude raise/lower mission or the phase change mission. The Perifocal reference frame was the simplest reference frame to implement. In the figure υ represents the true anomaly. The plane change mission, Mission 3, involved a 15 deg plane change; for that mission the Geocentric-Equatorial Reference frame was chosen to model the mission. A diagram of the Geocentric Equatorial Reference frame is shown Figure 7.

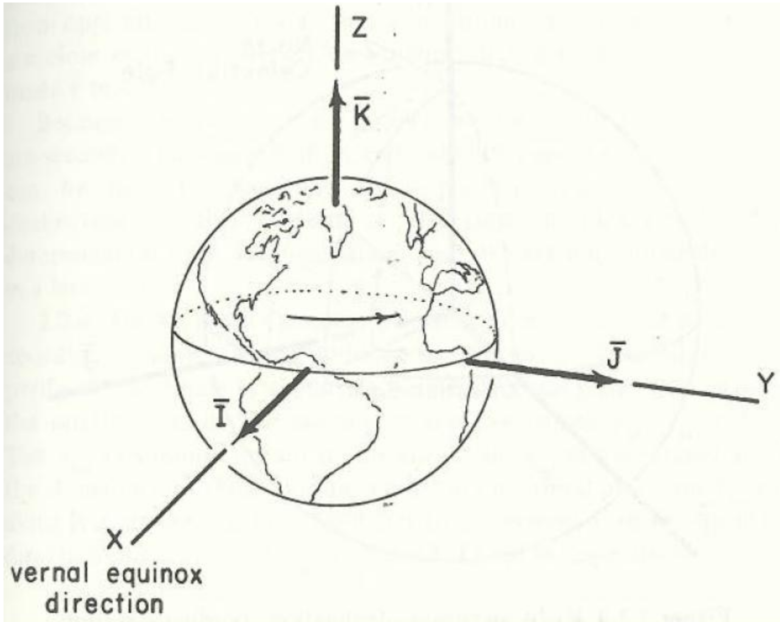

Figure 7: Geocentric-Equatorial Reference Frame[2]

As can be seen from Figure 7 the Geocentric-equatorial reference frame is well suited for a plane change maneuver. The Perifocal reference frame projected onto the Geocentric-Equatorial Reference frame is depicted below.

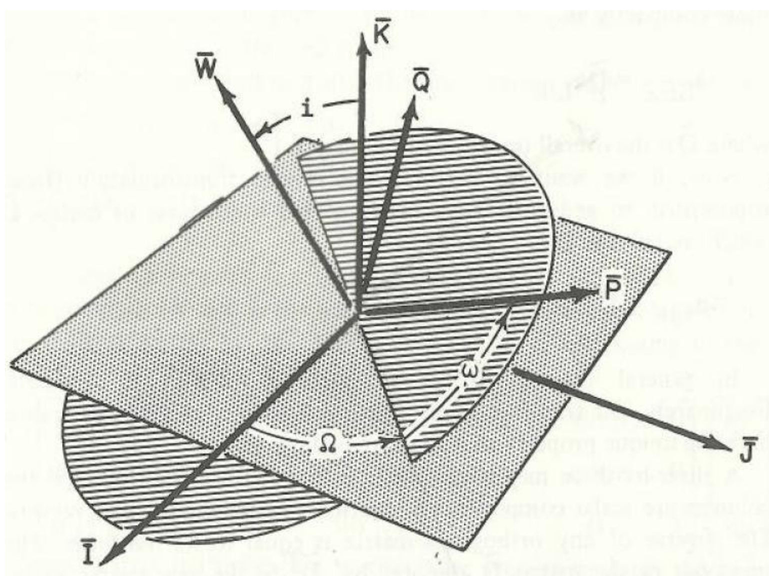

Figure 8: Perifocal and Geocentric-Equatorial Reference Frame[2]

The transfer between the perifocal and the geocentric-equatorial reference frame consists of performing a 3-1-3 rotation with the Right Ascension of the Ascending Node, the Argument of Perigee, and the inclination of the orbit. This transfer is discussed in the development of equations section for Mission 3. The process is outlined in Equation 18 through Equation 21.

Results

The results for each mission are described in detail in the following sections along with the development of equations. Each mission is discussed in its entirety before the next mission is discussed.

Mission 1: 12 hour 180 degree Phase Change and Return

To complete this mission the spacecraft was required to perform a 180 degree rephase maneuver within 12 hours and return to its initial orbit in 30 days. The chemical system performed the rephase maneuver via a Hohmann-type maneuver. The spacecraft entered a transfer orbit that caused it to re-enter the original orbit when the rephase angle was exactly 180 degrees. The electric propulsion system increased its orbit through a spiral maneuver then returned to the initial orbit when the 180 deg phase change was complete.

Mission 1 Development of Equations

To determine the best transfer orbit based on the time constraint the following equations were used:

$$(\boldsymbol{\omega_{init}} - \boldsymbol{\omega_{trans}})\mathbf{t_{trans}} = \boldsymbol{\theta}$$ **Equation 1**

Where ω_{init} is the rotation rate of the initial orbit and ω_{trans} is the rotation rate of the transfer orbit, t_{trans} is the total amount of time of the transfer and θ is the desired rephase angle.

$$\mathbf{t_{trans}} = \mathbf{n_{rot}}\mathbf{P_{trans}}$$ **Equation 2**

In Equation 2, n_{rot} represents the number of rotations of the transfer orbit and P_{trans} represents the period of the transfer orbit. Equation 2 was substituted back into Equation 1. The following well known equation for the period of an orbit was also used.

$$\mathbf{P_{init}} = \frac{\mathbf{2\pi}}{\boldsymbol{\omega_{init}}}$$ **Equation 3**

Equation 3 was rearranged solving for w_{init} and inserted into Equation 1 which resulted in Equation 4. The same equation as Equation 3 was used to solve for w_{trans} for the transfer orbit.

$$\left(\frac{\mathbf{2\pi}}{\mathbf{P_{init}}} - \frac{\mathbf{2\pi}}{\mathbf{P_{trans}}}\right)\mathbf{n_{rot}}\mathbf{P_{trans}} = \boldsymbol{\theta}$$ **Equation 4**

Solving for P_{trans} resulted in,

$$\mathbf{P_{trans}} = \left(\frac{\boldsymbol{\theta}}{\mathbf{2\pi n_{rot}}} + \mathbf{1}\right)\mathbf{P_{init}}$$ **Equation 5**

With Equation 5, the number of rotations n_{rot} was varied to find the transfer orbit which met the mission constraints. Since P_{init} is known based on the initial orbit as well as the rephase angle, the number of rotations, n_{rot}, may be changed to find a variety of transfer orbits. The effect of changing n_{rot} is discussed in the upcoming Mission 1 Results section.

To calculate the low thrust electric propulsion transfer orbit the resulting change in velocity was calculated with the following equation,

$$\Delta V = I_{sp} g_0 \log\left(\frac{m_{init}}{m_{current}}\right) \qquad \textbf{Equation 6}$$

The velocity of the spacecraft was updated with Equation 7. Since the change in velocity for an electric propulsion system is very small with a small time step and the change in velocity was applied perpendicular to the velocity vector the new orbit remained circular.

$$V^{i+1} = V^{i} \pm \Delta V \qquad \textbf{Equation 7}$$

The plus/minus symbol in Equation 7 was due to the orbit increasing or decreasing in size. This was changed based on the rephase angle completed and the necessary transfer time.

$$r^{i+1} = \frac{\mu}{\left(V^{i+1}\right)^2} \qquad \textbf{Equation 8}$$

The radius of the spacecraft was found with Equation 8. The period of the spacecraft was calculated with the well known equation for the period of a circular orbit,

$$P^{i+1} = 2\pi\sqrt{\frac{\left(r^{i+1}\right)^3}{\mu}} \qquad \textbf{Equation 9}$$

The new rotation rate was calculated with a rearranged Equation 3. The new true anomaly may be calculated with the following equation,

$$\eta^{i+1} = \eta^{i} + \omega^{i+1} t_{step} \qquad \textbf{Equation 10}$$

where t_{step} is the desired time step. The rephase angle may be calculated with the following equation,

$$\theta = \eta^{i+1}_{transfer} - \eta^{i+1}_{initial} \qquad \textbf{Equation 11}$$

Equation 11 was used to calculate the rephase angle resulting from the electric propulsion system.

Mission 1 Code Explanation

The MATLAB code developed to solve rephase missions was written such that the optimal n_{rot} value would be calculated based on a maximum trip time. The spacecraft mass, initial orbit parameters, time constraint, electric propulsion system values, chemical propulsion system values, and desired rephase amount were inputs. The code output the transfer time and propellant mass required for the case where only the chemical propulsion system performed the maneuver, as well as for the case where only the electric propulsion system performed the maneuver. This output reflected if the electric propulsion system was able to achieve the transfer within the time constraint.

The time constraint was applied for the chemical system alone, but it was not applied for the electric system alone because the electric propulsion system was unable to perform the change within the time constraint. The combined system values were also automatically generated, using the time left between the optimal chemical propulsion system transfer time and the time constraint. The code developed for this mission is included in Appendix A.

Mission 1 Results

An example of the chemical propulsion transfer orbit in the Perifocal reference frame is shown in Figure 9. The transfer orbit plotted represents the chemical transfer which would achieve the 180 degree rephase with the least amount of propellant required.

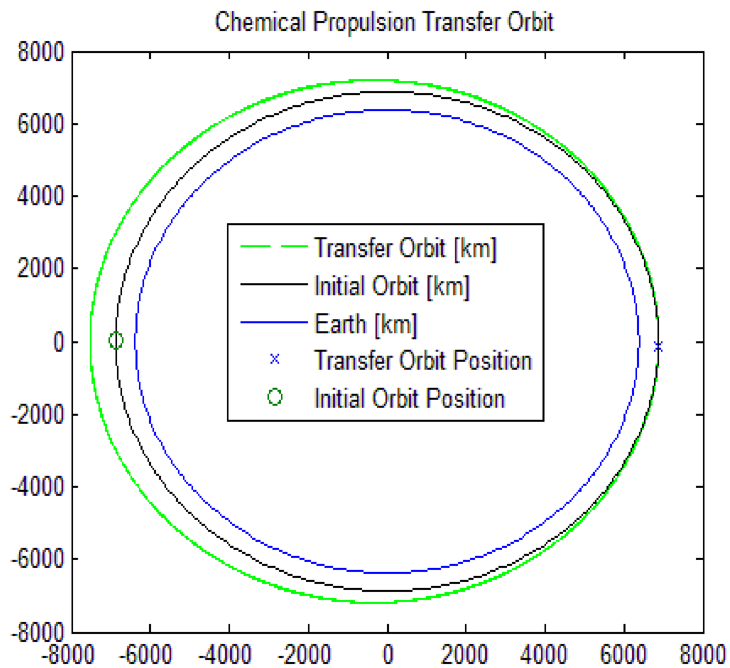

Figure 9: Mission 1 Chemical Propulsion Transfer Orbit

The "o" in Figure 9 represents the location of where the spacecraft would be had it remained in the initial orbit. The "x" represents the new position of the spacecraft resulting from the transfer orbit. The green dashed line represents the transfer orbit that resulted in the minimum propellant usage while still meeting the mission constraint of 12 hours. The total mission time required was 11.83 hours and 24.58 kg propellant for one 180 degree rephase maneuver. This transfer orbit was for 7 rotations (n_{rot}) values from Equation 5.

The required propellant mass and time for the electric propulsion and chemical propulsion systems are shown in Table 1 for a 180 degree rephase and return maneuver within the 12 hour time limit for each transfer.

Table 1: Mission 1 Values

	Mission Constraint	Electric Propulsion System	Chemical Propulsion System	Combined System
Transfer Time (hr)	12.00	46.96	11.83	12.00
Propellant Mass (kg)	< 80 kg	5.52	49.16	49.06
Delta V (m/s)	N/A	265.92	676.91	676.87

The values in Table 1 represent the entire mission with a 12 hour 180 degree rephase and a 30 day return. The combined system resulted in a propellant savings of around 0.1 kg. The mass savings was minimal because the time constraint of 12 hours prevented the electric propulsion system from performing much of the rephase maneuver. More time allotted to the chemical propulsion system enabled the spacecraft to transfer into a more optimal Hohmann transfer orbit which saved more propellant than thrusting with the electric propulsion system.

Once the optimal chemical transfer orbit was determined the electric propulsion system was modeled to thrust for the excess time available. That decreased the propellant mass only slightly because the electric propulsion system had a thrusting time of around 1 hour. The exact time that the electric propulsion system was allotted to thrust for may be seen in Figure 13 and discussed in more detail there.

The number of rotations , n_{rot}, used in the calculation of the transfer orbit had a substantial effect on the total propellant mass required by the chemical propulsion system to complete the maneuver. This is reflected in Equation 5; as n_{rot} increases, P_{trans}

decreases causing the transfer semi-major axis to become less and the period of the transfer orbit to become closer to P_{init}. As P_{trans} approaches P_{init} the delta-v required decreases. The relationship between time constraint and the propellant required for the chemical system to perform the maneuver may be seen in Figure 10.

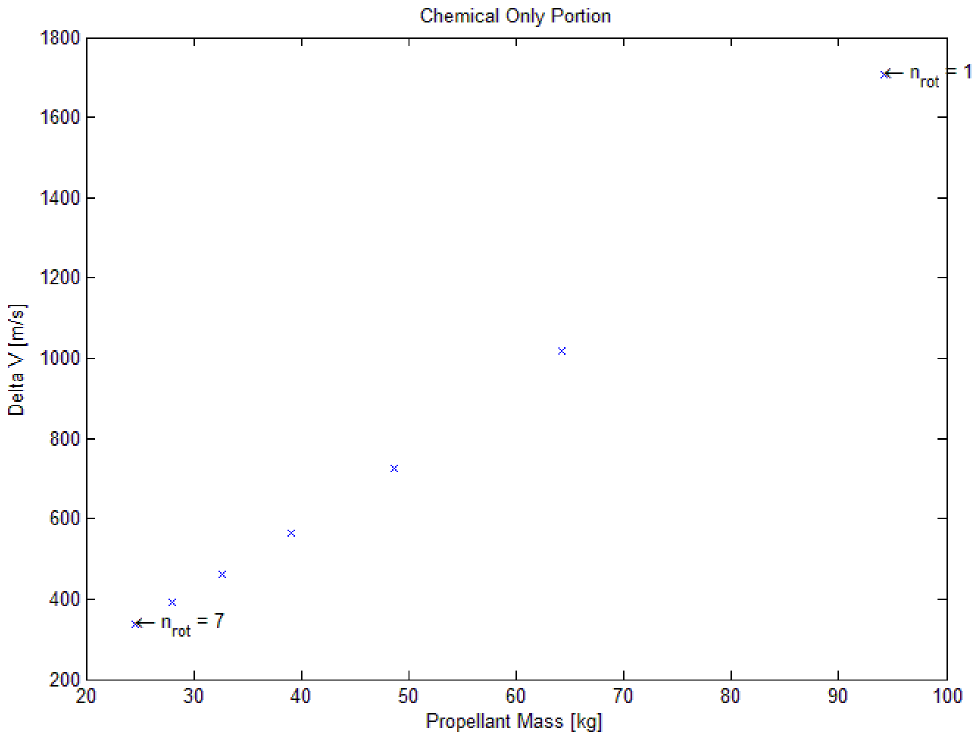

Figure 10: Mission 1 Effect of Number of Rotations on Propellant Mass

The point of least propellant mass, around 24 kg, corresponds to the largest number of rotations achievable to still meet the constraint of time and rephase angle. The upper-right point represents the least efficient transfer orbit. That transfer would achieve the phase change in the least time, but would exceed the mass constraint of 80 kg propellant. As the time constraint increased the trend line in Figure 10 would shift down and to the left because the number of rotations would increase.

The electric propulsion system performed a spiral orbit transfer to achieve the rephase amount. The electric propulsion system would thrust tangential to the orbit it was currently in. The new orbit after the burn was slightly larger or smaller depending on the mission than the previous orbit; the new orbit was assumed to be circular. This assumption was valid due to the small levels of thrust and small incremental change in altitude of the spacecraft. Figure 11 shows the transfer orbit using the spiral approach for the electric propulsion system.

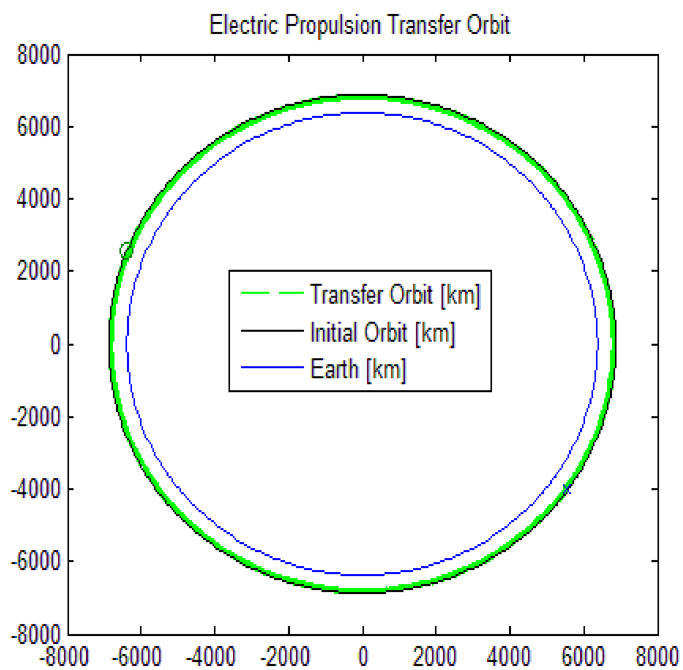

Figure 11: Mission 1 Electric Propulsion Transfer Orbit

The electric propulsion system alone exceeded the 12 hour time constraint. The "o" in Figure 11 represents the location of where the spacecraft would be had it remained in the initial orbit. The "x" represents the new position resulting from the transfer orbit.

The transfer orbit resulted in the rephase angle of 180 degrees being achieved, regardless of the time constraint, in Figure 11.

The gap between the amount of propellant required by the chemical system, 49.16 kg, and the propellant required by the electric propulsion system, 8.04 kg, provides a large trade space that may be explored. The next set of figures explores the mission space if the time constraint were larger than 12 hours. The time was increased from the 11.83 hr required by the chemical system to 59.76 hr required by the electric system. In coupling the chemical and electric propulsion systems the transfer time would increase but inversely the propellant mass would decrease.

The majority of the transfer maneuver, with a 12 hour time constraint, must be performed by the chemical system. Decreasing the number of rotations for the chemical propulsion system increases the amount of propellant required as may be seen in Figure 10. The largest number of rotations achievable in the time constraint should not be decreased; doing so greatly increases the required propellant as could be concluded by examining Equation 5.

A small trade space may be used which utilizes the time left between the total time the chemical propulsion system requires, and the time constraint. The electric propulsion system was modeled to thrust for a specified amount of time then the chemical propulsion system performed the remainder of the rephase maneuver to achieve the final constraints. Figure 12 shows the propellant mass savings which was calculated.

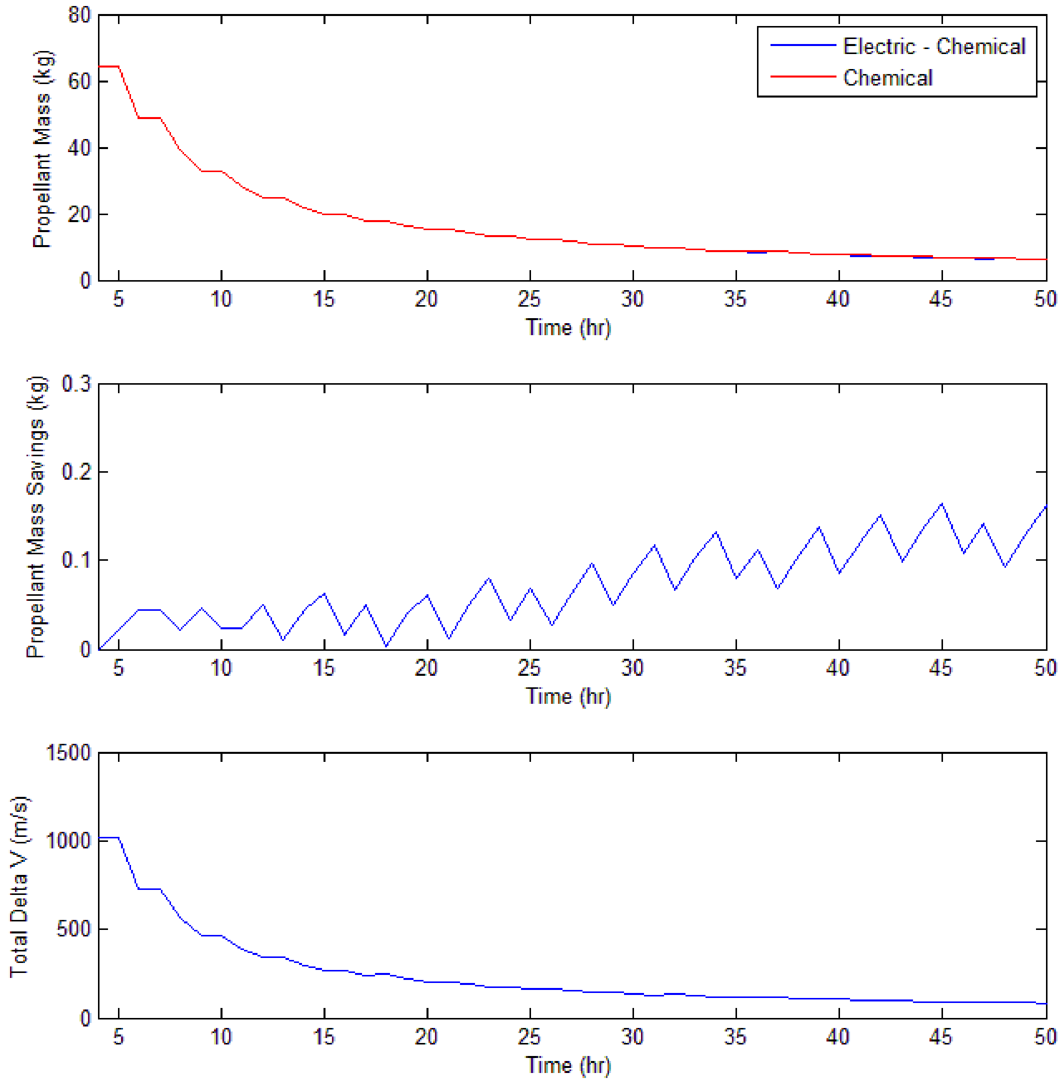

Figure 12: Mission 1 Potential Mass and Delta V Savings

Figure 12 shows the potential mass savings through the utilization of a multi-mode propulsion system for a variety of time constraints. The results portrayed in Figure 12 represent a mission that was performed primarily by a chemical propulsion system. On the top plot in Figure 12 the propellant mass value for a transfer time of 12 hours matches the value from Figure 10. The stepped appearance in Figure 12 results from the additional number of rotations, n_{rot} from Equation 5, which may be performed in the

additional time. A time constraint of 4 to 5 hours would result in 2 rotations possible. With an increasing trip time the rotation number would increase causing the transfer orbit to require less energy resulting in the propellant required to decrease. The effect of additional rotations on required propellant becomes smaller as seen from the smaller steps looking like a gradually sloped line.

The electric propulsion system thrusts such that the rephase amount of 180 degrees was achieved just prior to the time constraint. This is shown in Figure 13. The upper plot in Figure 12 shows the propellant mass required to perform a 180 degree rephase maneuver as well as the amount of propellant mass required for a combined chemical-electric, or multi-mode, system. The lines look very similar from this plot, but the difference between them is shown in the second plot.

The middle plot shows the propellant mass savings with a combined system verses the trip time constraint. For a trip time constraint of less than 4 hours there are no gains in having an electric propulsion system perform part of the maneuver because the amount of time that it would be able to thrust for was so minimal. The saw-tooth look results from the time step and the time the electric propulsion system was able to thrust for. This value may be seen in Figure 13; the saw-tooth look of this figured is reflected in Figure 12. The mass savings resulting from the electric propulsion system thrust for 0.1 hours would be less than the electric propulsion being able to thrust for 1.4 hours. The fluctuating time the electric propulsion was able to thrust for causes the saw-tooth look of the center plot. As the time constraint increased the propellant mass savings increased also the total delta-v required for the maneuver decreased as shown from the bottom plot in Figure 12.

As the trip time constraint was lengthened the chemical propulsion system was able to perform an additionally minimum delta-v Hohmann transfer orbit resulting in substantial mass savings.

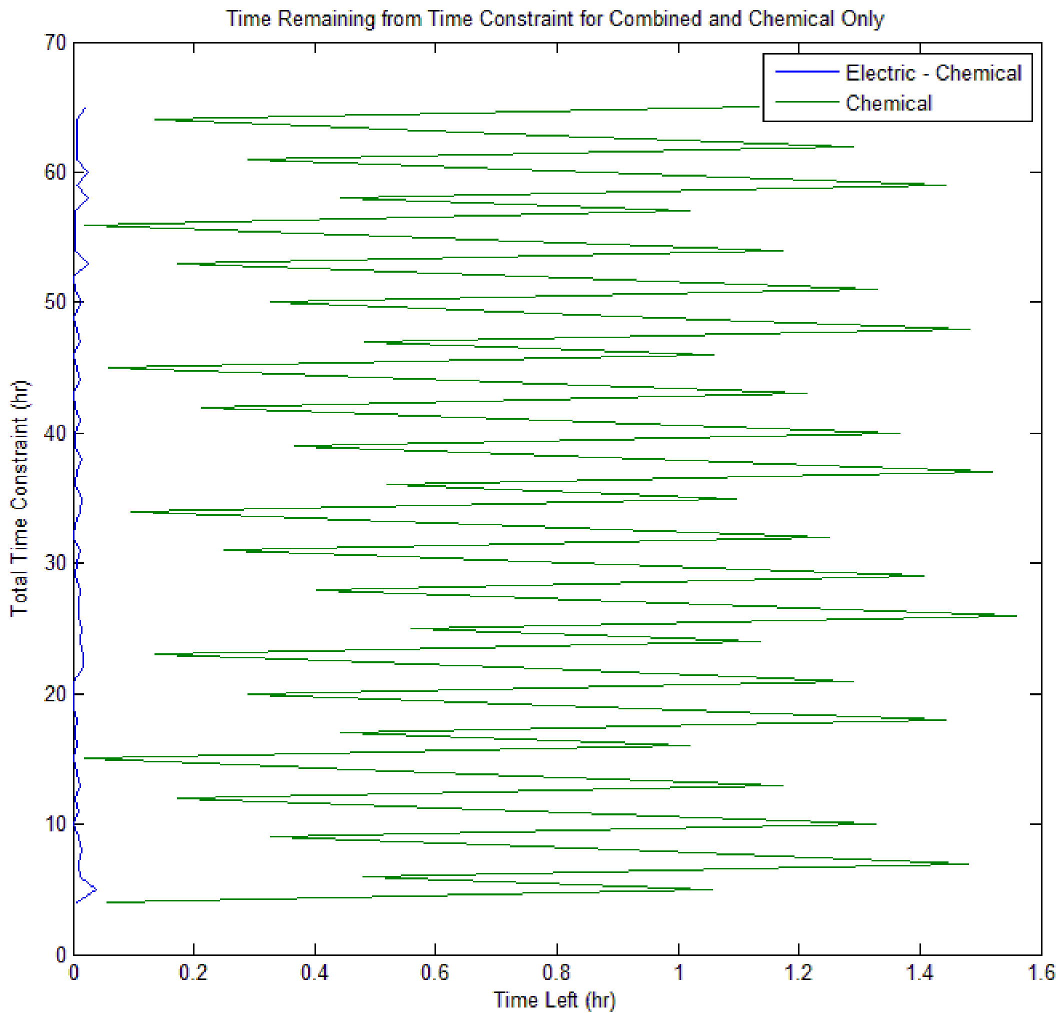

Figure 13: Mission 1 Time Constraint vs Time Left

Figure 13 displays the amount of time available for the electric propulsion system to thrust for. The trip time constraint was varied from about 4 hours to 65 hours to complete the rephase maneuver. Each time constraint resulted in a different Hohmann transfer orbit for the chemical propulsion system, which resulted in a different total trip

time. The total trip time was subtracted from the total time constraint which resulted in the green “chemical” propulsion line. That time was then used by the electric propulsion to thrust for; since the time available required fluctuated the mass savings also fluctuated with is seen in the jagged appearance in Figure 12.

When the electric propulsion system was utilized the propellant mass required decreased slightly and the time constraint was met within minutes. When the electric propulsion system was modeled to thrust for longer than the “time left” value in Figure 13 there was a substantial increase in the propellant required. That was due to the chemical system having to transfer into a less optimal transfer orbit to meet the rephase amount within the time constraint. Figure 14 shows the difference a longer time constraint has on the overall propellant mass and delta-v required to complete the 180 degree rephase maneuver.

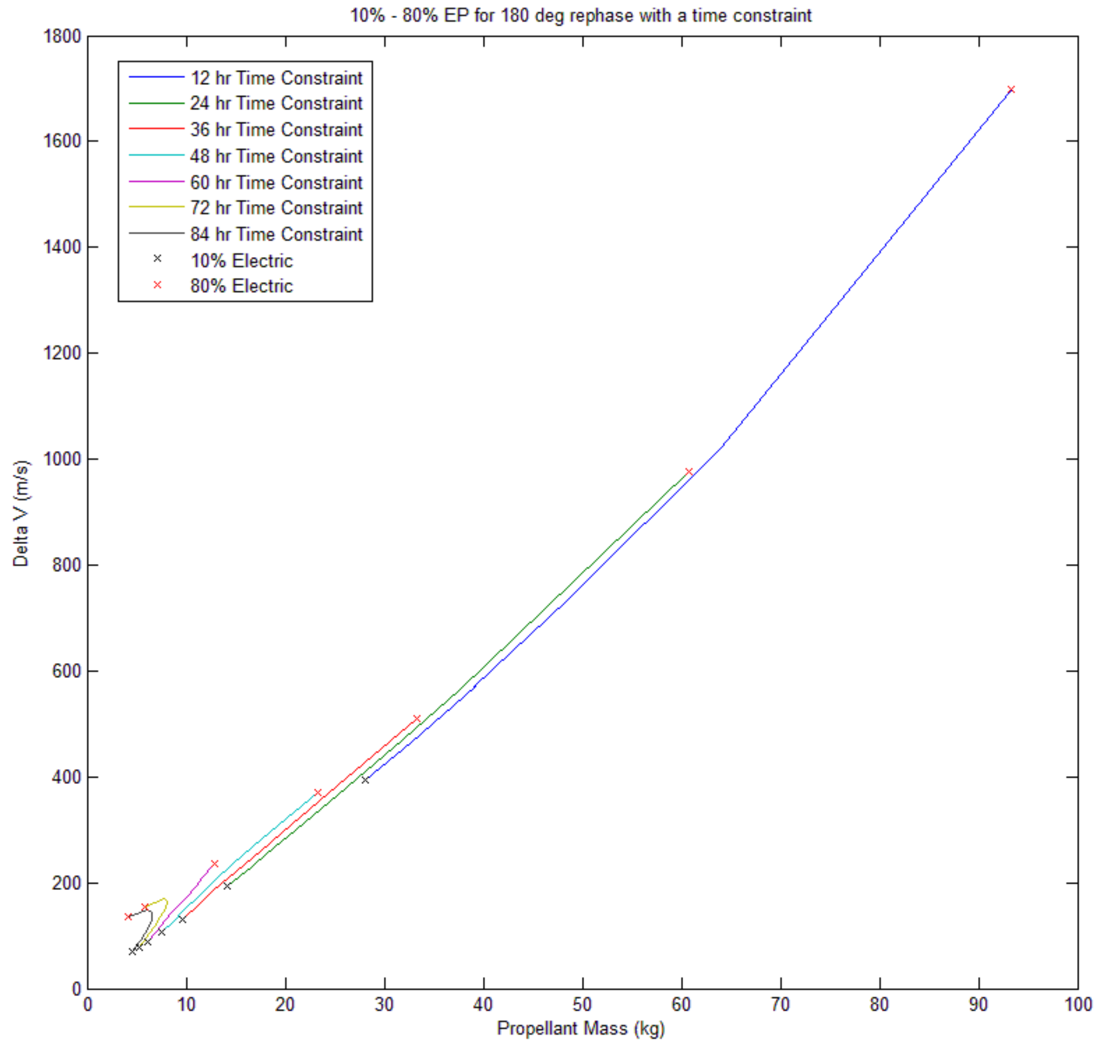

Figure 14: Mission 1 Delta V vs Propellant Mass with Varying Time Constraint

Figure 14 shows how the change in mission time constraint affected the total propellant and delta-v necessary to complete the 180 degree rephase mission. The 12 hour time constraint resulted in the largest amount of propellant. The propellant required decreased as the time constraint increased. The lines in Figure 14 represent a range of electric propulsion thrusting times that vary from the electric propulsion system thrusting for 10% of the total time, or 1.2 hours, to 80% of the total time, or 9.6 hours. The chemical system would utilize the remainder of the time to complete the rephase

maneuver. The 80% electric propulsion thrust profile resulted in more propellant than the 10% electric propulsion case until the time constraint was around 84 hours. This result was due to the chemical propulsion system having to transfer into a less optimal transfer orbit to achieve the rephase angle within the time constraint.

As the time constraint increased the range of propellant mass and delta-v required decreased. This can be seen from the length of the lines becoming shorter in Figure 14 as the time constrain increased. As the time constraint becomes larger the difference between the 80% and the 10% electric propulsion values decrease. A time constraint of 84 hr shows the 80% electric system using slightly less propellant than the 10% electric propulsion system. The 72 hour time constraint resulted in the primarily chemical system using slightly less propellant because 80% of 72 hours was just under the amount of time required by the electric propulsion system to complete the maneuver.

Rephase Angle Variations on Time Constraint

A separate analysis investigated different rephase angles and the overall time constraint. A large time constraint was applied to both the electric and chemical propulsion systems to perform a rephase maneuver.   The results may be seen in Figure 15.

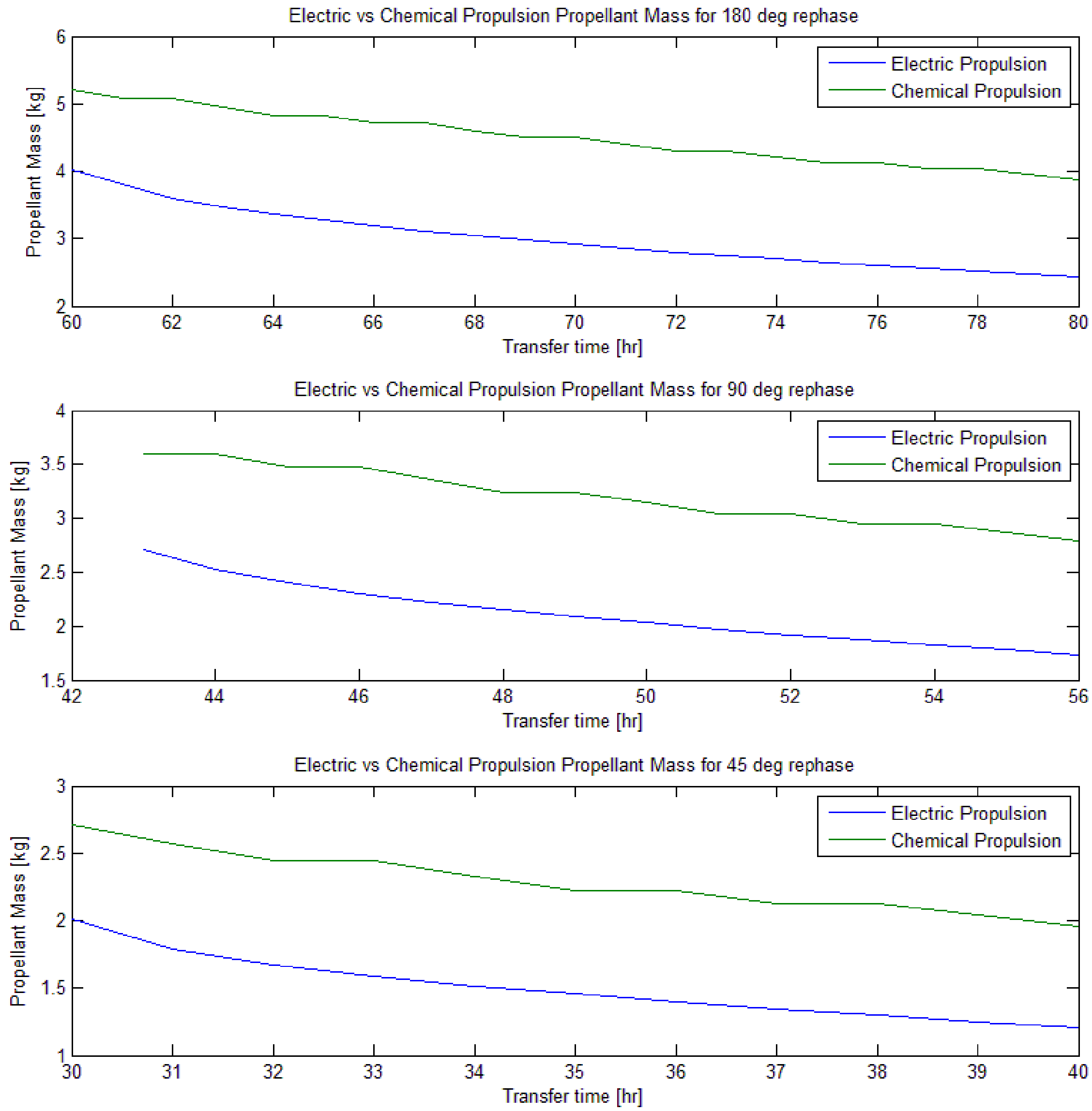

Figure 15: Mission 1 Large Time Constraint on EP and Chemical Propulsion

As shown from the plots included in Figure 15, both the chemical and electric propulsion systems benefit from an increasing transfer time limit. The chemical propulsion system can achieve the rephase angle with much less propellant than it initially could but still requires around 1 kg of additional propellant to achieve the rephase amount at longer time constraints than the electric propulsion system does.

Figure 15 shows the effect of a larger time constraint on three different rephase values: 180, 90, and 45 degrees. The different rephase amounts may be seen in the titles of the plots from Figure 15. The transfer time for the electric propulsion system begins at the point where it is able to complete the entire rephase maneuver on its own. The electric propulsion system thrusts constantly for the entire time constraint, unless it is able to thrust to a larger orbit then wait there until it thrusts to return to the initial orbit. The chemical propulsion system performs two thrust maneuvers: one go into a transfer orbit and a second to return to the initial orbit once the rephase amount has been achieved. For a 180 degree rephase angle and a transfer time constraint of 60 hours an electric propulsion system could potentially save over 1 kg of propellant; for this long duration that savings is around 25% of the total propellant required.

Ultimately, using electric propulsion for part of the time constraint will result in minor mass savings. Longer duration time constraints will result in additional mass savings. Varying the thrust profile decreases the necessary propellant mass slightly, but the largest driver for propellant mass required in a large rephase mission is the time constraint. This reflects how Multi-Mode Propulsion allows real-time optimization depending on required mission time or phase angle.

Mission 2: Orbit raise to 1500 km in 48 hours and return in 30 days

The spacecraft had a mass of 180 kg and began in a circular orbit at an altitude of 500 km. To complete this mission the satellite must perform a 1000 km altitude increase maneuver to achieve an altitude of 1500 km, within 48 hours and return to the initial orbit in 30 days. The 48 hour time constraint could not be achieved by using only the electric

propulsion system. The electric propulsion system modeled had a specific impulse of 600 sec and a thrust of 0.11 N. The electric propulsion system took 217.68 hours to complete the maneuver. This would require 14.64 kg of propellant and 499.49 m/s delta-v. The chemical system was modeled to perform the rephase maneuver by doing a Hohmann-type maneuver; it completed the altitude raise in 0.88 hour and required 35.03 kg of propellant. These values are shown in Table 2. The electric propulsion system was able to perform the return maneuver in the 30 day time constraint with the least amount of propellant so the following thrust techniques were applied only to the 48 hour time constraint orbit raising maneuver.

Several thrust variations for the combined system were modeled for this orbit transfer. The combinations varied the order and number of chemical and electric propulsion thrust segments, as well as the duration the chemical and electric was allotted to thrust for. The total time for the electric propulsion system to thrust for was varied based on the overall time constraint of 48 hrs. The time for the chemical system did not need to be varied because it was modeled as performing a Hohmann maneuver and the trip time was relatively constant at around 0.88 hr which was half the period of the transfer orbit. This fluctuated slightly depending on what the necessary altitude increase was. The electric propulsion system had the total time it could thrust for varied between 0% - 98% of the total time constraint. The electric propulsion system was not allowed to thrust for longer than 98% of the time constraint because the chemical system was required to perform the remaining altitude change in the remaining time. If the electric propulsion system would thrust for more than 98% of the time the overall time constraint would then be exceeded.

Mission 2 Development of Equations

The equations used to calculate the chemical transfer for an orbit raise maneuver were the well known equations associated with a Hohmann maneuver[2-5].

$$\mathbf{a_{trans} = \frac{r_{final} - r_{init}}{2}} \qquad \textbf{Equation 12}$$

With the initial and final orbit radius defined, the period and energy of the orbit can easily be calculated.

$$\mathbf{P_{trans} = 2\pi\sqrt{\frac{(a_{trans})^3}{\mu}}} \qquad \textbf{Equation 13}$$

$$\mathbf{E_{trans} = \frac{-\mu}{2a_{trans}}} \qquad \textbf{Equation 14}$$

The required change in velocity can then be calculated with the next group of equations.

$$\mathbf{V_{trans1} = \sqrt{2\left(\frac{\mu}{r_{final}} + E_{trans}\right)}}$$

$$\mathbf{V_{trans2} = \sqrt{2\left(\frac{\mu}{r_{final}} + E_{trans}\right)}} \qquad \textbf{Equation 15}$$

$$\mathbf{\Delta V_1 = V_{init} - V_{trans1}}$$

$$\mathbf{\Delta V_2 = V_{final} - V_{trans1}}$$

$$\mathbf{\Delta V = \Delta V_1 + \Delta V_2} \qquad \textbf{Equation 16}$$

The delta-v resulting from the Equation 16 represents the total delta-v required to change orbits from the initial to the final orbit. Similar equations may be used to calculate the electric propulsion transfer as were previously stated in the Mission 1 Development of Equations section.

The electric propulsion system can raise the altitude of the orbit in a spiral transfer through the use of the following equations.

$$\dot{\mathbf{m}} = \frac{\mathbf{T}}{\mathbf{I_{sp} g_0}} \qquad \textbf{Equation 17}$$

$$\mathbf{m^{i+1} = m^i - \dot{m}\Delta t} \qquad \textbf{Equation 18}$$

$$\mathbf{\Delta V^{i+1} = Ispg_0 log\left(\frac{m^i}{m^{i+1}}\right)} \qquad \textbf{Equation 19}$$

$$\mathbf{V^{i+1} = V^i \pm \Delta V^{i+1}} \qquad \textbf{Equation 20}$$

The new velocity, from Equation 20, is plus or minus depending on the direction of the applied thrust which is a result of the orbit being increased or decreased.

$$\mathbf{r^{i+1}} = \frac{\boldsymbol{\mu}}{\left(\mathbf{V^{i+1}}\right)^2} \qquad \textbf{Equation 21}$$

If the orbit is circular then,

$$\mathbf{a^{i+1} = r^{i+1}} \qquad \textbf{Equation 22}$$

And as such the new period can be calculated utilizing Equation 13.

Mission 2 Code Development

Three main MATLAB codes were developed to analyze this mission. The first code was a chemical propulsion altitude change code. The inputs were initial altitude, final altitude, spacecraft mass, propulsion system characteristics, and trip time constraints. The chemical propulsion system was modeled utilizing the most efficient altitude change maneuver, a Hohmann transfer. The transfer orbit was calculated and the spacecraft's position, required delta-v, trip time, and propellant required were outputs for an impulsive burn.

The second code developed modeled the electric propulsion transfer. The electric propulsion code had the same inputs as the chemical propulsion system but with

available power as an additional input. The available power was not incorporated into the transfer. It was assumed that the power available was constant and provided the specific impulse of 600 sec and thrust of 0.11 N. The delta-v was applied tangentially and the updated orbital elements were calculated. The thrust was modeled as being continuous for the time allotment.

The third MATLAB script that was created utilized both the chemical and electric propulsion codes and varied the initial inputs based on the percent time allotment for each propulsion system. The chemical and electric propulsion codes were run numerous times until the final orbit corresponded with the indicated final orbit conditions with the thrust profile specified. The MATLAB codes written to model Mission 2 are included in Appendix B.

Mission 2 Results

The combined systems analyzed were: "Electric – Chemical", "Electric – Chemical – Electric", "Electric – Chemical – Electric remaining time", "Chemical – Electric", and "Chemical – Electric – Chemical" for the orbit raise maneuver in the 48 hr time constraint. Additional descriptions of the combined systems and their orbit profiles are below.

The combined system values listed in Table 2 represent the "Electric – Chemical – Electric" system. This system involved the electric propulsion system thrusting for 49% of the time constraint, the chemical propulsion performing a Hohmann maneuver, then the EP system thrusting for another 49% of the time constraint to reach the destination orbit. The profile of that transfer orbit is represented in Figure 20. The best

combined system, alongside the electric and chemical propulsion systems performing the transfer alone are shown in Table 2.

Table 2: Mission 2 Values

	Mission Constraint	**Electric Propulsion System**	**Chemical Propulsion System**	**Combined System**
Transfer Time (hr)	48.00	217.68	0.88	47.97
Propellant Mass (kg)	< 80 kg	14.64	35.03	30.24
Delta V (m/s)	N/A	499.49	498.98	488.97

The combined system, the "Electric – Chemical – Electric" profile was only one of numerous thrust profiles modeled. Figure 16 shows the propellant and change in velocity required for a variety of thrust profiles. The "Electric – Chemical – Electric" maneuver required the least amount of propellant of the various thrust profiles. The propellant mass listed for the mission constraint represents the total propellant mass available for the orbit raise and return. Using only the chemical propulsion system to complete the entire maneuver would require 70.06 kg of propellant but would achieve the maneuver quickly. Using only the electric propulsion system would exceed the 48 hour transfer limit, but would result in twice the amount of remaining propellant for future maneuvers.

Utilizing the combined multi-mode propulsion system would result in a savings of around 5 kg for the orbit raise maneuver. Figure 16 displays the various thrust profiles which were analyzed.

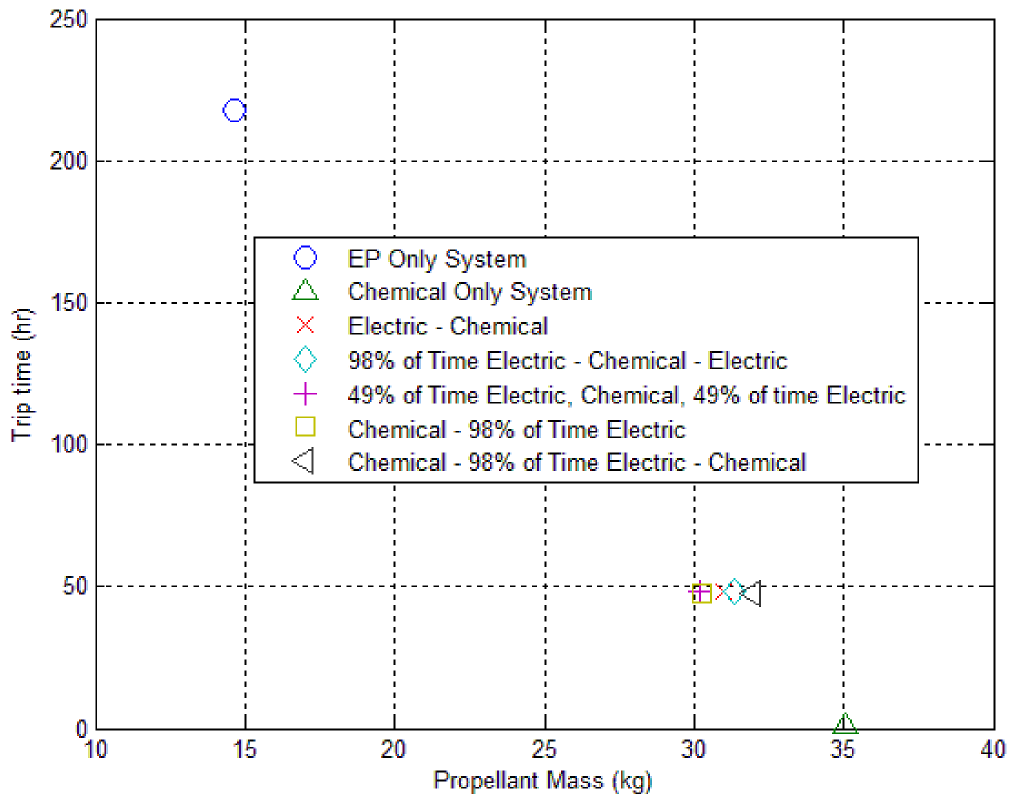

Figure 16: Mission 2 Propellant Mass vs Trip Time for Thrust Variations

All the data points in Figure 16 represent separate thrust profiles analyzed as compared against the pure electric system and the pure chemical system. All the profiles were run for a variety of thrust times for the electric propulsion system. The points shown in the figure represent the least amount of propellant resulting from the various thrust profiles. The thrust profiles all require close to the 48 hour time constraint. If the time constraint were longer the thrust profiles would gradually move upward and to the left in the figure until they reached the electric propulsion system only point. The electric

propulsion system only point would move upward and to the left with increasing specific impulse. The specific impulse used for the electric propulsion system was 600 sec. The thrust profiles with a varying trip time ranging about 1 hour to 48 hours is shown in Figure 17.

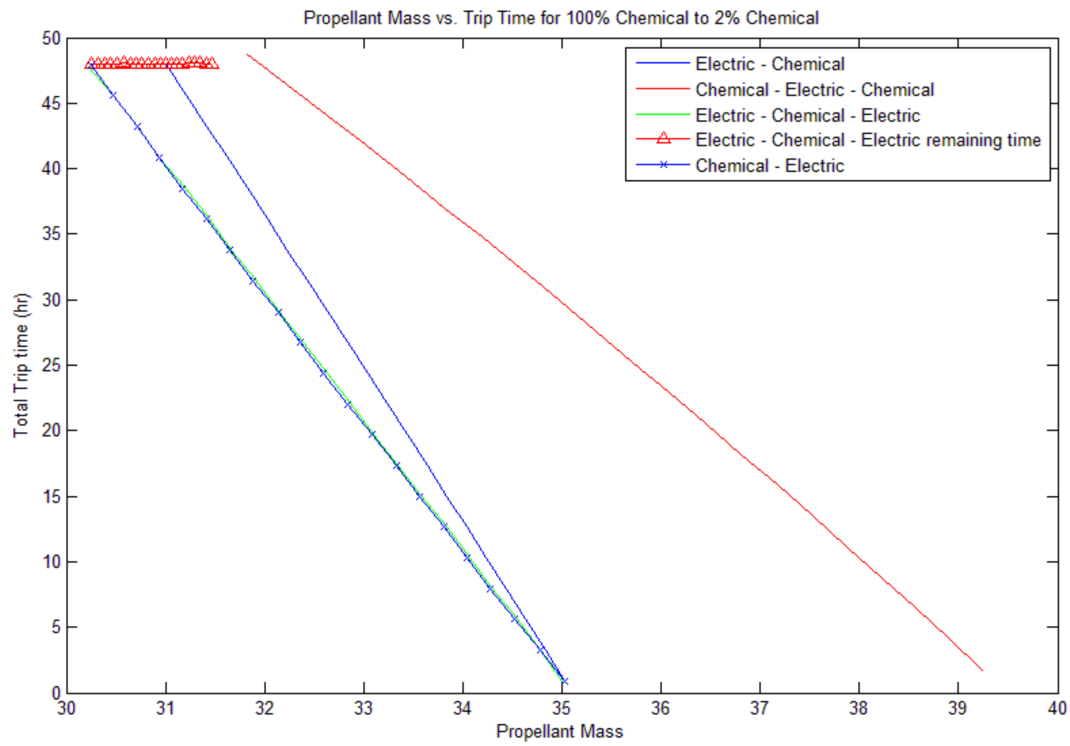

Figure 17: Mission 2 Thrust Profiles vs Propellant Mass

Each thrust profile was modeled with the electric propulsion system thrusting for 0% - 98% of the total time constraint which resulted in the electric propulsion system thrusting for around 47 hours. All of the profile lines originate in the lower right hand corner which shows the total propellant mass decreasing as the trip time increases. The only exception was in the “Electric – Chemical – Electric remaining time” line represented by the triangles. This line begins at the top left of Figure 17 for the 0% of the

total time constraint for the electric propulsion system.   This is discussed in more detail in the upcoming “Electric – Chemical – Electric remaining time” section.

Of the five separate thrust profiles shown in Figure 17, the “Electric – Chemical – Electric”, “Chemical – Electric”, and the “Electric – Chemical – Electric remaining time” provide the lowest propellant mass required. The “Electric – Chemical – Electric” line in Figure 17 appears nearly identical to the “Chemical – Electric” line. The descriptions of these maneuvers are described below.

All the thrust profiles below meet the mission constraints of mass and time. The maximum time the electric propulsion was allowed to thrust for was 98% of the time constraint which was around 47 hours. This was because the chemical propulsion system took half an orbit to complete its Hohmann transfer maneuver to reach the final orbit within the time constraint. All the profiles listed had a maximum electric propulsion thrust time of 47 hours and a minimum thrust time of 0 hours which would be an entirely chemical propulsion transfer.

Electric – Chemical

This thrust profile involved the electric propulsion system performing the first impulse ranging from 0 hours to 47 hours. A diagram of this thrust profile may be seen in Figure 18.

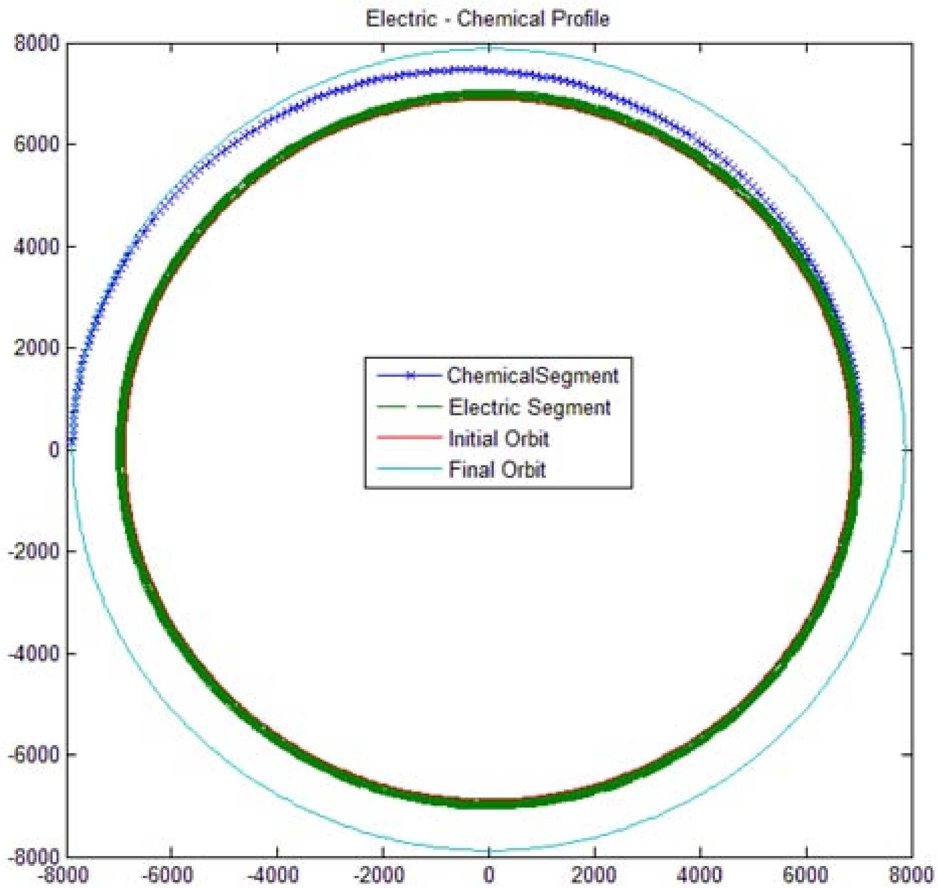

Figure 18: Mission 2 Electric – Chemical Profile

The amount of time the electric propulsion system was allotted to thrust for was varied from 0% of the total time to 98% of the total time, 0 – 47 hours, with the final altitude achieved by the electric propulsion system acting as the initial altitude for the chemical Hohmann maneuver. Figure 18 displays the orbit for the maximum time case for the electric propulsion system. As the electric propulsion system thrusts for a larger percent of the time, the total propellant mass required decreases. At the final time constraint where the electric propulsion system thrusts for 47 hours, the altitude changes from 6878 km – 7070.66 km, which is an altitude change of 192.66 km. The electric propulsion transfer portion causes the orbit to remain circular and the chemical

propulsion system to perform a smaller altitude raise. The propellant mass required for this thrust profile was 31.00 kg. The system values are listed in Table 3.

Chemical – Electric – Chemical

This thrust profile was analyzed with the electric propulsion system using 0% - 98% of the total time. The altitude achieved in the specified burn time by the electric propulsion system was used as a starting point for an iteration to determine what altitude increase the chemical system would need to provide. Allowing the electric propulsion system to thrust for 0 hours resulted in the least optimal transfer because the chemical system would perform 2 separate Hohmann maneuvers to achieve the final orbit. The remaining altitude change to complete the maneuver was split between the two chemical propulsion segments. Each chemical propulsion segment performed roughly the same altitude increase. The electric propulsion system performed the altitude increase in between the chemical propulsion Hohmann maneuver segments. This thrust profile is shown in Figure 19.

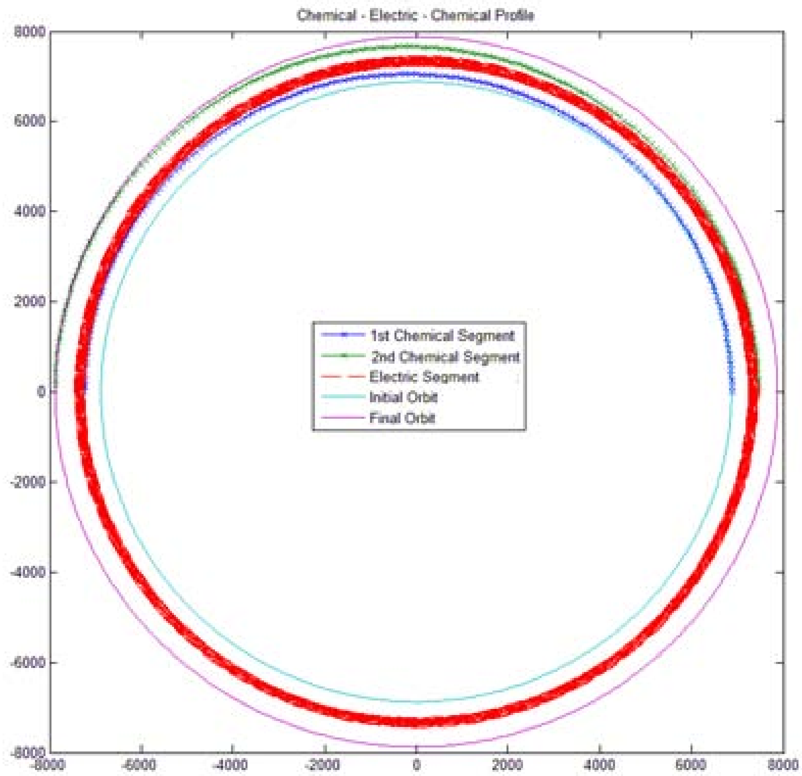

Figure 19: Mission 2 Chemical – Electric – Chemical Profile

Figure 19 shows the mission profile for the maximum time electric propulsion case. As seen in the figure this thrust profile involved the chemical propulsion system performing a Hohmann transfer to a specified altitude, the electric propulsion performing a spiral transfer for a specified amount of time, and the chemical propulsion system performing another Hohmann transfer to achieve the final altitude. The minimum propellant mass required using this thrust profile was 31.82 kg. The system values are listed alongside the other thrust profile values in Table 3. This thrust profile resulted in the largest amount of propellant required of the combined systems analyzed.

Electric – Chemical – Electric

This thrust profile was run with the electric propulsion system using 0% - 98% of the total time. For the 0% electric propulsion case, the propellant mass was the same as the chemical propulsion only maneuver. The time the electric propulsion system would thrust for was varied between 0 % to 49% for each electric propulsion segment before and after the chemical portion. The profile of this maneuver is shown below in Figure 20.

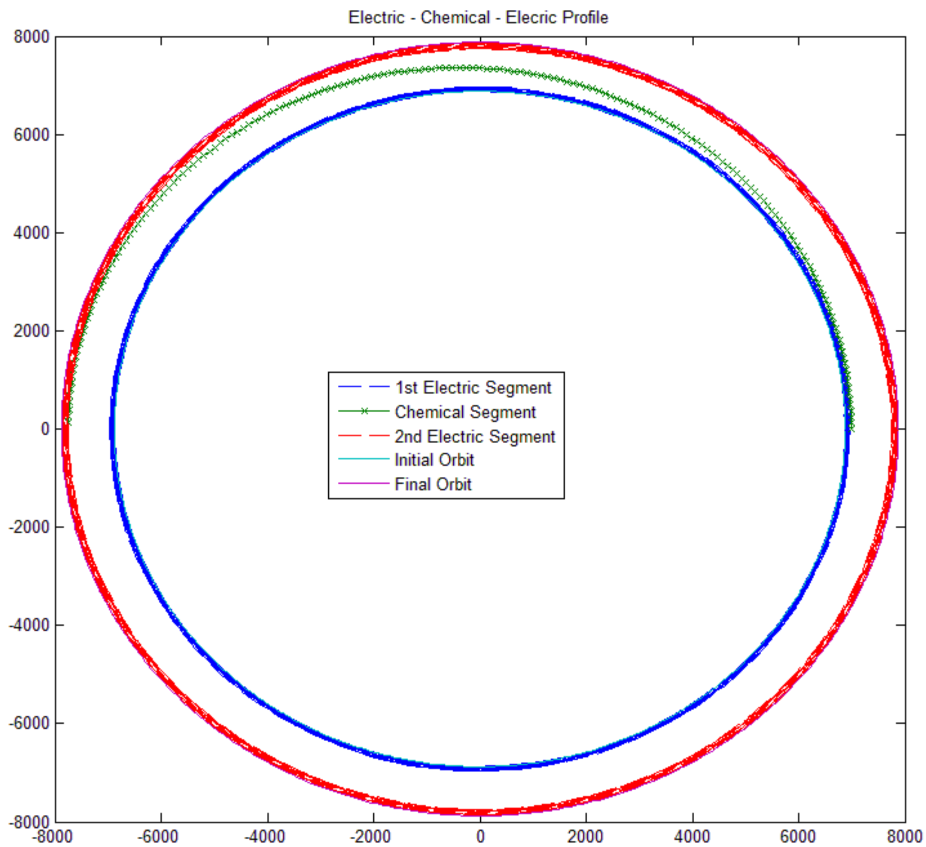

Figure 20: Mission 2 Electric – Chemical – Electric Profile

Figure 20 shows the mission profile for the electric propulsion system thrusting for a total of 47 hours. An iterative process was used such that electric propulsion system would thrust for half of the total allotted time, then the chemical system would perform a

Hohmann maneuver and increased the altitude enough such that the final electric propulsion segment could attain the final altitude of 7878 km within the amount of time allotted. The system values are listed alongside the other thrust profile values in Table 3. This thrust profile resulted in the least propellant required of all the combined systems modeled.

The thrust profile shown in Figure 20 displays the difference in utilizing the electric propulsion system at lower altitudes. The blue portion representing the first electric propulsion segment performs less of the altitude increase than the red line representing the second altitude increase maneuver though they are thrusting for the same amount of time. The electric propulsion system was able to perform more of the altitude change at a higher altitude due to the increasing delta-v resulting from the decreasing overall spacecraft mass.

Electric – Chemical – Electric remaining time

This thrust profile had the initial electric propulsion system thrust for 0 % - 98% of time requirement. The electric system would thrust for a specified amount of time then the chemical system would perform a Hohmann maneuver followed by the electric propulsion system thrusting up until the time constraint of 48 hr. This thrust profile was a slight modification on the "Electric – Chemical – Electric" profile.

The achieved altitude was calculated iteratively. The altitude the chemical system was required to attain was adjusted such that the 2nd electric propulsion thrust duration achieved the altitude within the time constraint. The minimum propellant mass required was at the initial case of 0% electric propulsion, which resulted in roughly the same

propellant mass as the “Chemical – Electric” thrust profile where the electric propulsion was thrusting for 98% of the time. The reason they didn’t achieve the same results was because the electric propulsion system the “remaining time” profile was allowed to thrust for slightly longer. This thrust profile required the least propellant due to the electric propulsion system providing a larger altitude increase when utilized later in the trajectory when the spacecraft had a lower overall mass. The orbit profile for this mission is shown in Figure 21.

98% Time **0% Time**

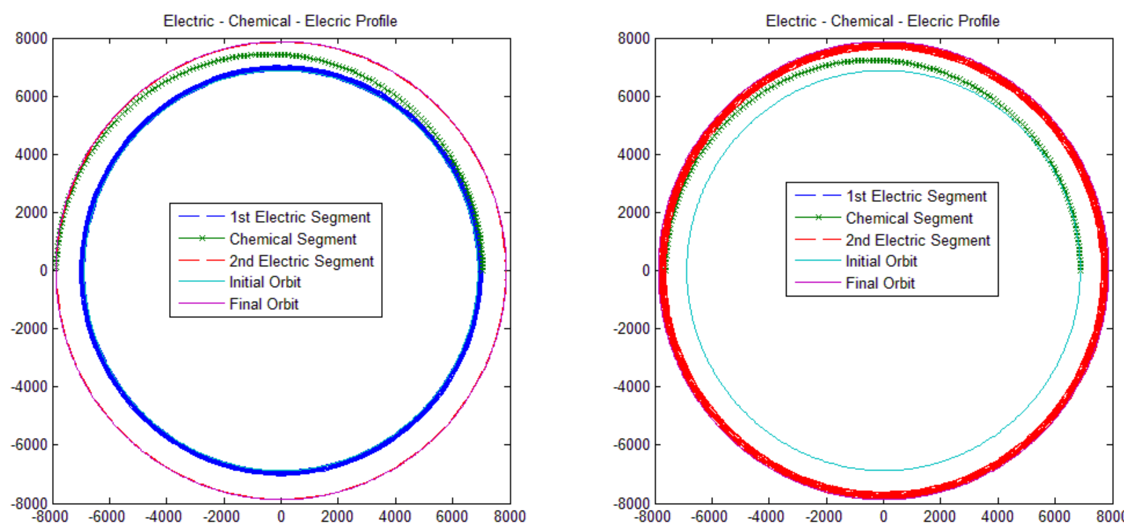

Figure 21: Mission 2 Electric – Chemical – Electric remaining time Profile

The plot on the left in Figure 21 displays the case where the electric propulsion system initially thrusts for 98% of the total time constraint. The plot on the right displays the case where there is no initial electric propulsion and the majority of the maneuver is completed by the chemical propulsion system, then the electric. It can be noted from these plots that in the right hand plot the electric propulsion system performs a greater amount of the altitude change maneuver. This results in the right hand case of 0%

electric propulsion having a lower required propellant to complete the maneuver. The result for the 0% electric propulsion thrust duration and the 98% electric thrust duration are listed alongside the other thrust profile values in Table 3.

Chemical – Electric

The “Chemical – Electric” thrust profile had the chemical propulsion system thrust first then the electric propulsion system would thrust for a specified amount of time. This thrust profile is shown in Figure 22.

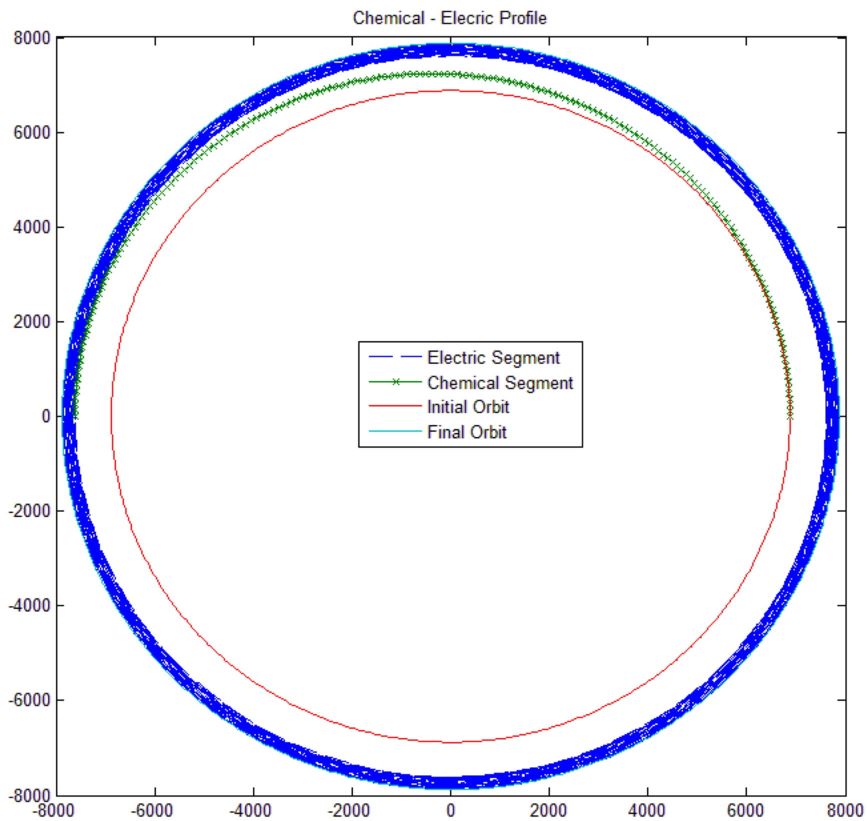

Figure 22: Mission 2 Chemical – Electric Profile

As can be seen from Figure 22, the chemical propulsion system initially performed a Hohmann maneuver then the electric propulsion system performed the

remainder of the mission with a spiral transfer. The calculation of the altitude that the chemical propulsion system must transfer to was completed iteratively to meet the mission constraints. This thrust profile required 30.25 kg of propellant mass to achieve the 1000 km altitude increase. The system values are listed alongside the other thrust profile values in Table 3.

Table 3: Mission 2 Thrust Profile Values

	Electric – Chemical	**Chemical – Electric – Chemical**	**Electric – Chemical – Electric**	**Electric – Chemical – Electric remaining time**	**Chemical – Electric**
Transfer Time (hr)	47.99	47.90	47.97	48.00 (47.98)	47.95
Propellant Mass (kg)	31.00	31.98	30.24	31.35 (30.25)	30.25
Delta V (m/s)	499.26	492.77	488.57	498.88 (499.17)	499.14
Electric Propulsion Altitude Change (km)	192.66	226.45	247.73	193.09 (266.19)	266.06

Some interesting items to note from Table 3 are that the "Electric – Chemical – Electric remaining time" has two separate numbers listed. These represent the two different plots shown in Figure 21, for the two separate cases of 98% of the time electric propulsion, which are the first column of numbers and the 0% of time electric propulsion which are the second column of numbers. The 0% of time electric propulsion results in the thrust profile to be almost the same as a "Chemical – Electric" profile. It is not exactly the same because the amount of time that the electric propulsion system was modeled to thrust for is not the same.

The best thrust profile appears to one where an electric impulse is before and after the chemical impulse. This had a very similar propellant usage as was required by a "Chemical – Electric" profile. Having the electric impulse after the chemical impulse enables the electric system to perform more of the altitude change maneuver. Having the chemical system perform the altitude change maneuver starting at a lower altitude appears to require slightly more propellant causing the "Electric – Chemical – Electric" thrust profile to use slightly less propellant than the "Chemical – Electric" case.

Figure 23 compares the average thrust against the propellant mass and trip time for the electric propulsion thrusting from 0% - 98% of the total time constraint.

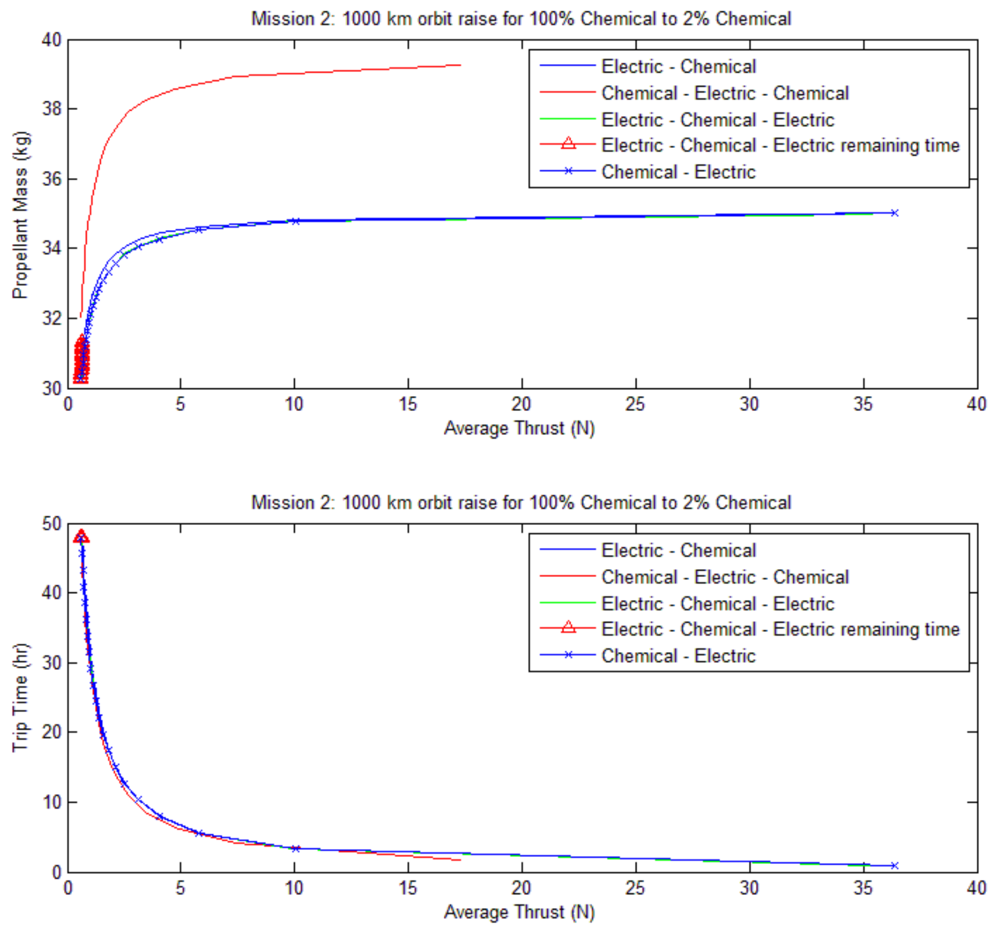

Figure 23: Mission 2 Average Thrust for Multiple Thrust Profiles

As may be seen in Figure 23 the largest amount of propellant was required by the "Chemical – Electric – Chemical" thrust profile. The rest of the thrust profiles appear almost identical for the various thrusting schemes. The "Electric – Chemical – Electric remaining time" was different than the other profiles because the trip time is always around 48 hr. From the lower plot in Figure 23 it may be seen that all the points overlap for that thrust profile; that was because the full time constraint was utilized by this system as opposed to the other thrust profiles. Lower average thrust results in lower propellant required and longer trip time.

Looking at the "Electric – Chemical – Electric remaining time" profile alone concludes that using chemical thrust then electric thrust is better than using electric thrust then chemical thrust. The least amount of propellant resulted in the thrust profile with the chemical situated between two equivalent electric propulsion segments. The effect of changing the percent electric thrust duration on the propellant required for the different thrust profiles is shown in Figure 24.

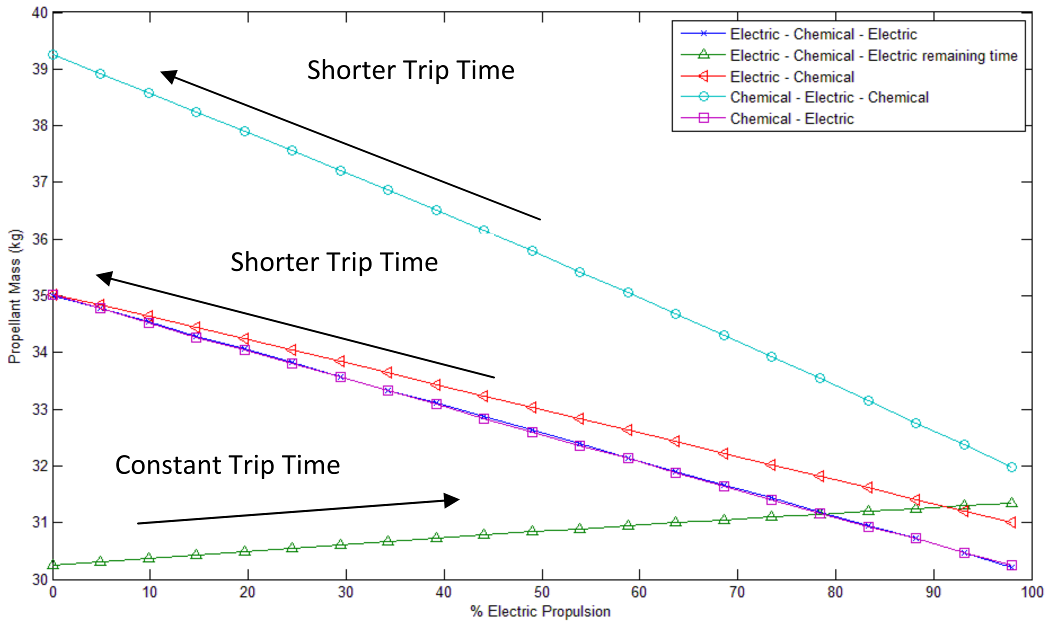

Figure 24: Mission 2 Effect of Percent Electric Propulsion on Propellant Mass

In general the propellant mass required decreases as the percent electric propulsion increases. This is switched for the "Electric – Chemical – Electric remaining time" case. The arrows indicate that for all thrust profiles except the "Electric – Chemical – Electric remaining time" case the trip time decreases as going towards the left hand side of the plot.

For a 1000 km altitude change maneuver with a 48 hour time constraint, utilizing a multi-mode propulsion system can decrease the propellant mass required from 35 kg to 30 kg. The entire mission with an altitude raise performed, and a return in 30 days would require around 70 kg of propellant for a chemical system to perform alone. With a combined system the total propellant used would be reduced to 44.85 kg which is a decrease of 36% in propellant use.

Mission 3: 15 degree Plane Change with Time Constraint

The spacecraft initially had a mass of 180 kg with 80 kg available for propellant. The spacecraft was located in a circular orbit at an altitude of 500 km. The destination orbit was at an altitude of 500 km and circular with at an inclination of 15 deg. The plane change was required to be completed in less than 90 days. The thrust vector direction was calculated such that the transfer orbit remained essentially circular.

The initial orbit and the final orbit with an inclination of 15 degrees are depicted in Figure 25 in the Geocentric Equatorial reference frame.

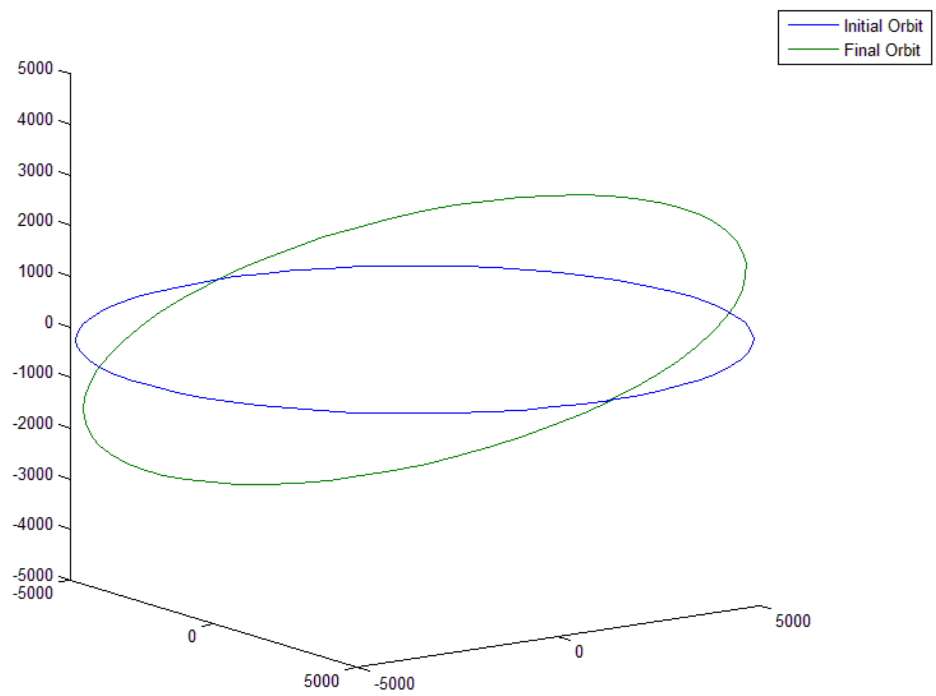

Figure 25: Mission 3 Geocentric Equatorial Reference Frame

Figure 25 shows a scaled version of the initial and final orbit in the Geocentric-Equatorial reference frame and the two crossover points that occur. The electric propulsion system was modeled initially to thrust at the intersection (crossover) locations. The crossover point was where the chemical propulsion system performed the cranking maneuver.

Mission 3 Development of Equations

For a plane change maneuver the perifocal reference frame could not be the only reference frame utilized. The Geocentric Equatorial Reference frame is better suited for plane change maneuvers. The method to be followed to perform an impulsive plane change is shown next.

The velocity vector of the spacecraft was broken into a transverse and a radial component[4,5,6,7].

$$\overrightarrow{\mathbf{V_1}} = \mathbf{V_{r1}}\hat{\mathbf{u}}_{\mathbf{r1}} + \mathbf{V_{\perp 1}}\hat{\mathbf{u}}_{\perp\mathbf{1}} \qquad \textbf{Equation 23}$$

$$\overrightarrow{\mathbf{V_2}} = \mathbf{V_{r2}}\hat{\mathbf{u}}_{\mathbf{r2}} + \mathbf{V_{\perp 2}}\hat{\mathbf{u}}_{\perp\mathbf{2}} \qquad \textbf{Equation 24}$$

In the equations above the "1" denotes the initial orbit and the "2" denotes the final orbit. The total change in velocity to transfer from the initial orbit to the final orbit may be calculated by,

$$\Delta\overrightarrow{\mathbf{V}} = \overrightarrow{\mathbf{V_2}} - \overrightarrow{\mathbf{V_1}} \qquad \textbf{Equation 25}$$

The delta-v was also calculated with the same equation as Equation 25, except with the velocity in the final orbit being in time step ahead of the velocity vector in the initial orbit. Doing that resulted in more propellant, so Equation 25 was utilized.

$$\Delta\overrightarrow{\mathbf{V}} = \mathbf{V_{r2}}\hat{\mathbf{u}}_{\mathbf{r2}} + \mathbf{V_{\perp 2}}\hat{\mathbf{u}}_{\perp\mathbf{2}} - \mathbf{V_{r1}}\hat{\mathbf{u}}_{\mathbf{r1}} - \mathbf{V_{\perp 1}}\hat{\mathbf{u}}_{\perp\mathbf{1}} \qquad \textbf{Equation 26}$$

$$\Delta\mathbf{V^2} = \Delta\overrightarrow{\mathbf{V}} \cdot \Delta\overrightarrow{\mathbf{V}} \qquad \textbf{Equation 27}$$

Multiplying the above equations and utilizing trigonometry identities resulted in:

$$\Delta\mathbf{V} = \Delta\mathbf{V} = \sqrt{(\mathbf{V_{r2}} - \mathbf{V_{r1}})^2 + \mathbf{V_{\perp 2}}^2 - \mathbf{V_{\perp 1}}^2 - 2\mathbf{V_{\perp 1}}\mathbf{V_{\perp 2}}\mathbf{cos}(\boldsymbol{\delta})} \qquad \textbf{Equation 28}$$

Where δ represents the inclination change between the two orbits. To perform an impulsive delta-v maneuver the spacecraft must thrust at the line of intersection between the initial orbit and the desired orbit; this is also referred to throughout the remainder of the paper as the crossover point. Performing the plane change at the apoapsis results in the radial component of velocity becoming zero such that the transverse portion of velocity can be treated as the actual velocity. Applying the aforementioned simplifications resulted in the following equation:

$$\Delta\mathbf{V} = \sqrt{\mathbf{V_2^2} + \mathbf{V_1^2} - 2\mathbf{V_2}\mathbf{V_1}\mathbf{cos}(\boldsymbol{\delta})} \qquad \textbf{Equation 29}$$

Since the initial and final orbits for this mission were the same except for the difference in inclination the equation simplifies further to become,

$$\mathbf{\Delta V = 2Vsin\left(\frac{\delta}{2}\right)} \qquad \textbf{Equation 30}$$

Since the thrust is applied at the crossover nodes V1 = V2 = V, which is a well known equation for plane change[1,2]. This equation is intended for a large impulse, which excludes an electric propulsion transfer. The resulting delta-v from this equation was used as if it were applicable for an electric propulsion system to compare the resulting propellant mass with more exact calculations.

The equation used to calculate the propellant required based on delta-v is:

$$\mathbf{m_p = m_i\left(1 - e^{\left(\frac{-\Delta V}{I_{sp}g}\right)}\right)} \qquad \textbf{Equation 31}$$

This equation resulted in 103.99 kg of propellant required for the chemical propulsion system which exceeded the propellant available for the maneuver. Utilizing this same equation for an electric propulsion system resulted in 51.52 kg of propellant required for the higher specific impulse of the electric propulsion mode.

The electric propulsion system was modeled by transferring the spacecraft position and velocity vector between the Perifocal and the Geocentric-Equatorial reference frame. Since the altitude of the initial circular orbit was known the following equation was used to transfer the known altitude into the position in the Perifocal reference frame[1].

$$\mathbf{\vec{r} = rcos(\eta)\hat{p} + rsin(\eta)\hat{q}} \qquad \textbf{Equation 32}$$

The velocity may similarly be transferred. In the equations η represents the true anomaly. In the equation below "p" is the semi-latus rectum, for a circular orbit this is equal to the magnitude of the position vector.

$$\vec{\mathbf{v}} = \left(\sqrt{\frac{\mu}{p}}\right)\left[-\sin(\eta)\hat{\mathbf{p}} + \left(\mathbf{e} + \cos(\eta)\right)\hat{\mathbf{q}}\right] \qquad \textbf{Equation 33}$$

These vectors were transferred into the Geocentric-Equatorial reference frame through the use of the following rotation matrices.

$$\vec{\mathbf{r}}_{\mathbf{ijk}} = \vec{\mathbf{r}}_{\mathbf{pqw}}\mathbf{R_3}(\boldsymbol{\Omega})\mathbf{R_1}(\boldsymbol{\pi} - \mathbf{i})\mathbf{R_3}(\boldsymbol{\omega}) \qquad \textbf{Equation 34}$$

Where,

$$\mathbf{R_3}(\boldsymbol{\omega}) = \begin{bmatrix} \cos(\omega) & \sin(\omega) & 0 \\ -\sin(\omega) & \cos(\omega) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \textbf{Equation 35}$$

$$\mathbf{R_1}(\boldsymbol{\pi} - \mathbf{i}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\pi - i) & \sin(\pi - i) \\ 0 & -\sin(\pi - i) & \cos(\pi - i) \end{bmatrix} \qquad \textbf{Equation 36}$$

$$\mathbf{R_3}(\boldsymbol{\Omega}) = \begin{bmatrix} \cos(\Omega) & \sin(\Omega) & 0 \\ -\sin(\Omega) & \cos(\Omega) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \textbf{Equation 37}$$

The same process was used to transform the velocity vector. The direction of the necessary velocity vector may be calculated from this new form of the velocity vector.

$$\vec{\mathbf{v}}_{\mathbf{ijk\,diff}} = \vec{\mathbf{v}}_{\mathbf{ijk\,final}} - \vec{\mathbf{v}}_{\mathbf{ijk\,transfer}} \qquad \textbf{Equation 38}$$

The velocity vector was made non-dimensional with the following equation,

$$\vec{\mathbf{v}}_{\mathbf{diff}} = \frac{\vec{\mathbf{v}}_{\mathbf{ijk\,diff}}}{\left|\vec{\mathbf{v}}_{\mathbf{ijk\,diff}}\right|} \qquad \textbf{Equation 39}$$

The straight brackets represent the magnitude of the vector. The change in velocity depends on the level of thrust (T) and the mass of the spacecraft (m_{sc}) as shown in the equations below.

$$\Delta\mathbf{v} = \int \frac{\mathbf{T}}{\mathbf{m_{sc}}}\mathbf{dt} \xRightarrow{\text{Constant Thrust}} \mathbf{T}\int \frac{\mathbf{dt}}{\mathbf{m_{sc}}} \xRightarrow{\text{small time step}} \frac{\mathbf{Tt_{step}}}{\mathbf{m_{sc}}}$$

Equation 40

If thrust, T, is assumed to be constant it may be taken out of the integral. Also, if the time step and change in mass of the spacecraft is small it may approximated outside of the integral.

The change in velocity vector was calculated as shown below.

$$\overrightarrow{\mathbf{\Delta v}} = \mathbf{\Delta v} * \vec{\mathbf{v}}_{\mathbf{diff}} \qquad \textbf{Equation 41}$$

The equation above was used to update the spacecraft's position and velocity vector.

$$\vec{\mathbf{r}}_{\mathbf{ijk}}{}^{\mathbf{i+1}} = \vec{\mathbf{r}}_{\mathbf{ijk}}{}^{\mathbf{i}} + \overrightarrow{\vec{\mathbf{v}}_{\mathbf{ijk}}}^{\mathbf{i}} * \mathbf{t}_{\mathbf{burn}} \qquad \textbf{Equation 42}$$

$$\vec{\mathbf{v}}_{\mathbf{ijk}}{}^{\mathbf{i+1}} = \vec{\mathbf{v}}_{\mathbf{ijk}}{}^{\mathbf{i}} + \vec{\mathbf{g}}_{\mathbf{ijk}}{}^{\mathbf{i}}\mathbf{t} + \overrightarrow{\mathbf{\Delta v}} \qquad \textbf{Equation 43}$$

Equation 43 and 44 were used to update the spacecraft state vector. The updated state was based on the previous state, as well as the effect of gravity and thrust. Gravity was initially calculated in the perifocal reference frame then transferred into the Geocentric-Equatorial reference frame through a 3-1-3 rotation; the same as shown for the position in Equation 35.

Mission 3 Code Development

The development of the MATLAB scripts for Mission 3 required the largest amount of time. They involved transferring between the Perifocal and the Geocentric-Equatorial reference frame. The initial orbit of the spacecraft was modeled to incorporate the effects of gravity on the spacecraft causing it to naturally "fall" around the earth. The state vector was composed of the position and velocity vectors and was updated each time step with the effect of gravity and any thrust that occurred. The direction of the thrust was described in the development of equations section. If-then statements were

utilized to turn on an off the thrust. When the spacecraft was not thrusting the state vector was updated based only on the effects of gravity.

The position was updated by transferring the updated state from the Geocentric-Equatorial reference frame to the Perifocal reference frame. The true anomaly was tracked to ensure the updated position in one reference frame corresponded with the correct location in the other reference frame. The position was checked to ensure that the transfer orbit remained circular. Step size in MATLAB had a large effect on the orbit remaining circular. If the state vector was not checked then the orbit calculated throughout a single orbit with no thrust did not remain exactly circular. The step size had to be less than a few seconds for iteration error to not build up. If the step size were larger than a few seconds one orbit would become slightly non-circular. That effect would not have much effect but in a low thrust transfer over 90 days the orbit would become much larger than it was supposed to be. The delta-v would have a gradually lessoning effect on the change in inclination. To ensure the thrust profile changed the inclination only the state vector was updated in the Geocentric-Equatorial reference frame to determine the change in inclination then the updated position was checked against the position in the new Perifocal reference frame, based on the true anomaly, to ensure that the transfer orbit was not growing in size.

Mission 3 Results

Several analysis methods were applied to compare the difference in trip time and required propellant mass that resulted from incrementally, versus instantaneously, updating the change in velocity and position of the spacecraft. The second part of the

analysis compared various thrust durations for the electric propulsion system with the chemical propulsion system performing the remainder of the plane change when the time constraint was reached.

The change in velocity required by the chemical system resulted in a delta-v of 1987.31 m/s, from Equation 14, with 103.99 kg of propellant, from Equation 15. The electric propulsion system was analyzed with several different thrust schemes and analysis techniques. The first method was to treat the change in velocity over a period of time as though it had an instantaneous effect on the velocity of the spacecraft. That method of analysis was compared with:

1) Updating the spacecraft's state vector every minute for a period of 15 min beginning at the crossover point and,

2) Centered on the crossover point and updated every few seconds, and

3) Thrust continuously.

These three methods are described next.

Using these analysis techniques the spacecraft was able to reach the necessary 15 degree plane change within the time constraint for some of the thrust profiles. The results for all these methods and additional trade studies are described in Table 4 through Table 6 and in the following sections. The transfer orbit for all the electric propulsion thrust profiles had the same general flight profile shown in Figure 26.

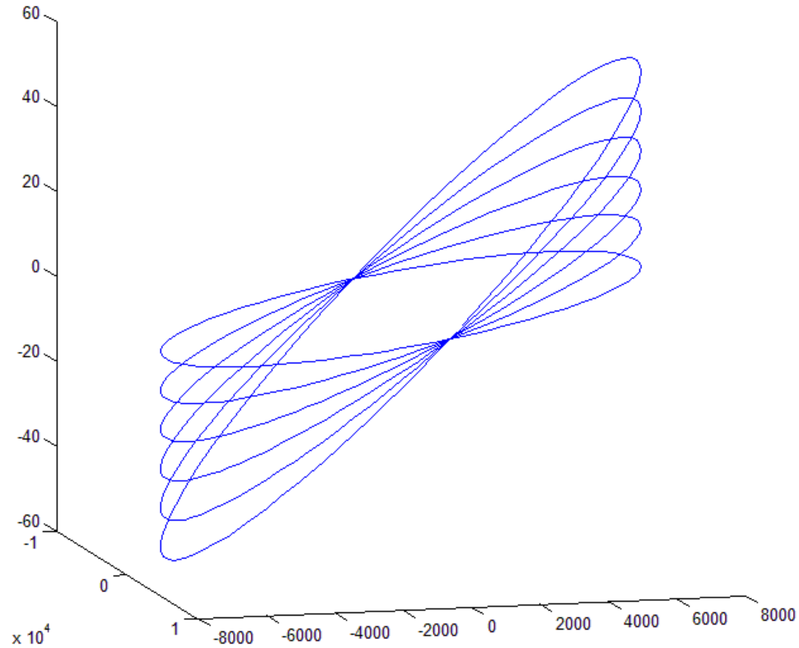

Figure 26: Mission 3 Generic Flight Profile

Figure 26 depicts an electric propulsion transfer orbit. The thrust vector between the transfer orbit and the final orbit were calculated each orbit prior to applying the increase in velocity from the electric propulsion system. Doing this caused the transfer orbit to change inclination slightly, but not change the other orbital parameters. The crossover points were the same for each orbit during the transfer orbit.

Single Burn Profile

The position and velocity of the spacecraft during the transfer orbit were updated instantaneously, incrementally, and continuously. The instantaneous change in position was the least realistic of all the orbit transfers modeled. The location of the instantaneous thrust implementation may be seen in Figure 27.

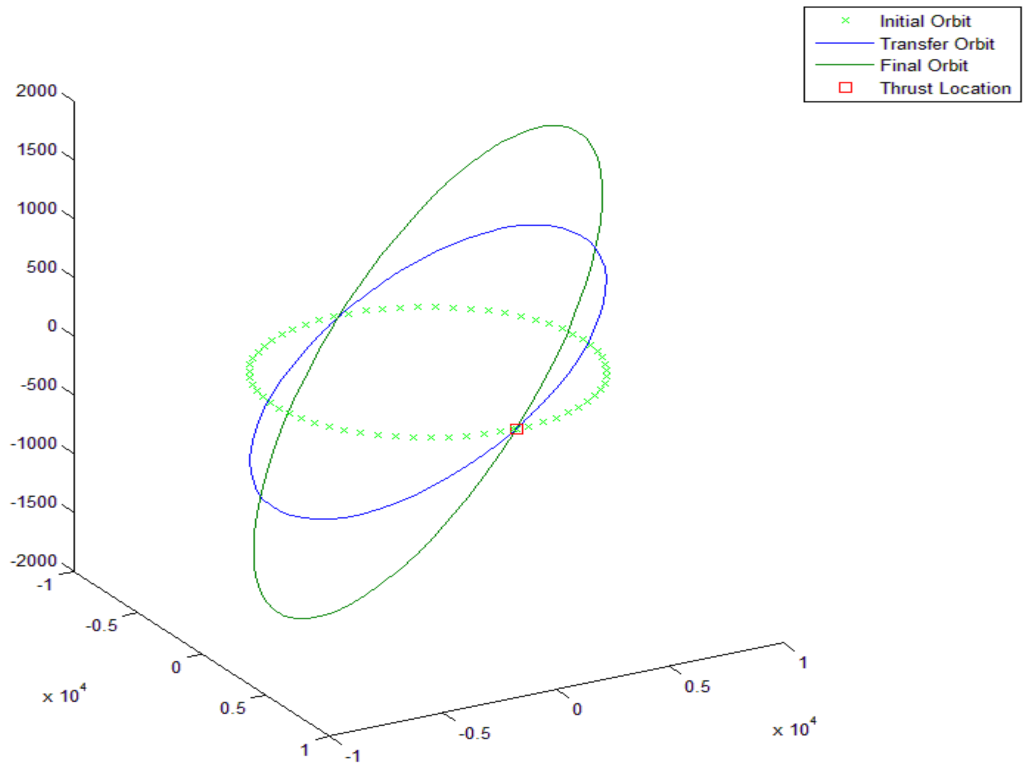

Figure 27: Mission 3 Electric Propulsion System One Impulsive Burn

The red square represents the location where the change in velocity from the 15 min thrust was applied; this method reflects that the electric propulsion system has much more thrust than it actually has. The effect of the change in velocity on the position and velocity vector were applied instantaneously at the crossover point. When viewed from the “-k” direction in the Geocentric-Equatorial reference frame the orbits appear circular; the elliptical look resulted from the axis of the plot. One 15 min burn achieved about half the desired plane change. This analysis method was compared with two other orbit updating techniques described next.

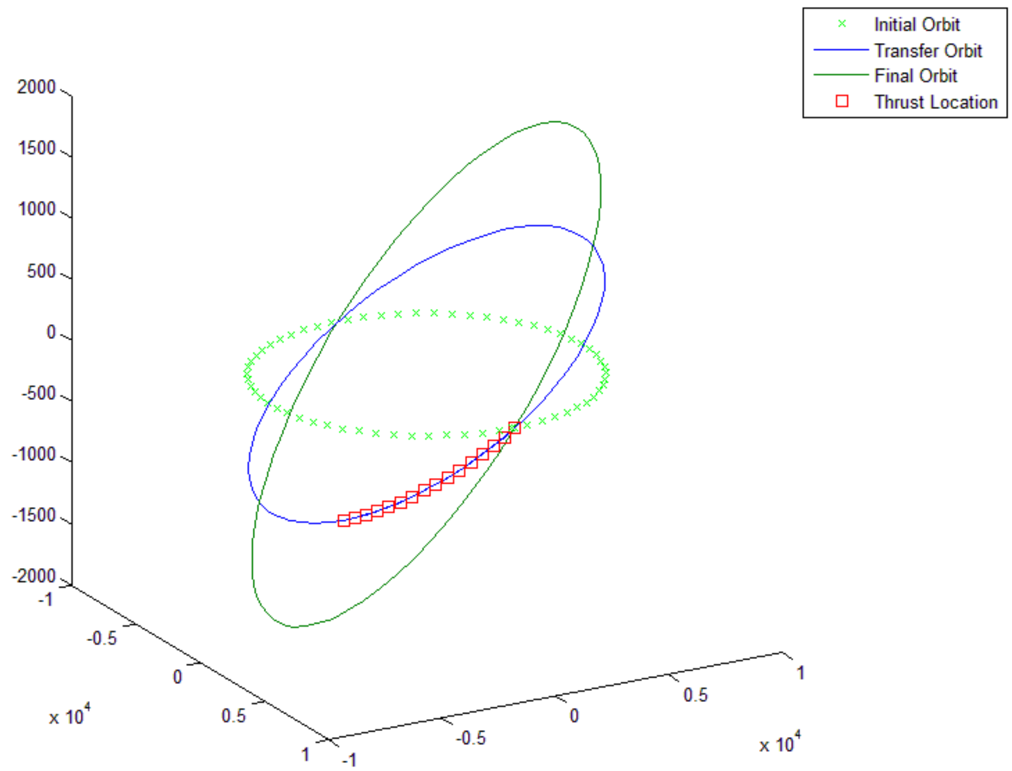

Figure 28: Mission 3 One Burn Beginning at Crossover Point

The thrust profile depicted in Figure 28 involved the transfer spacecraft thrusting each orbit beginning at the crossover point and lasting for 15 min. The spacecraft's position and velocity vector were updated continuously during the thrust portion. The results for this thrust profile are shown in Table 4. A slight variation on this thrust profile is shown in Figure 29.

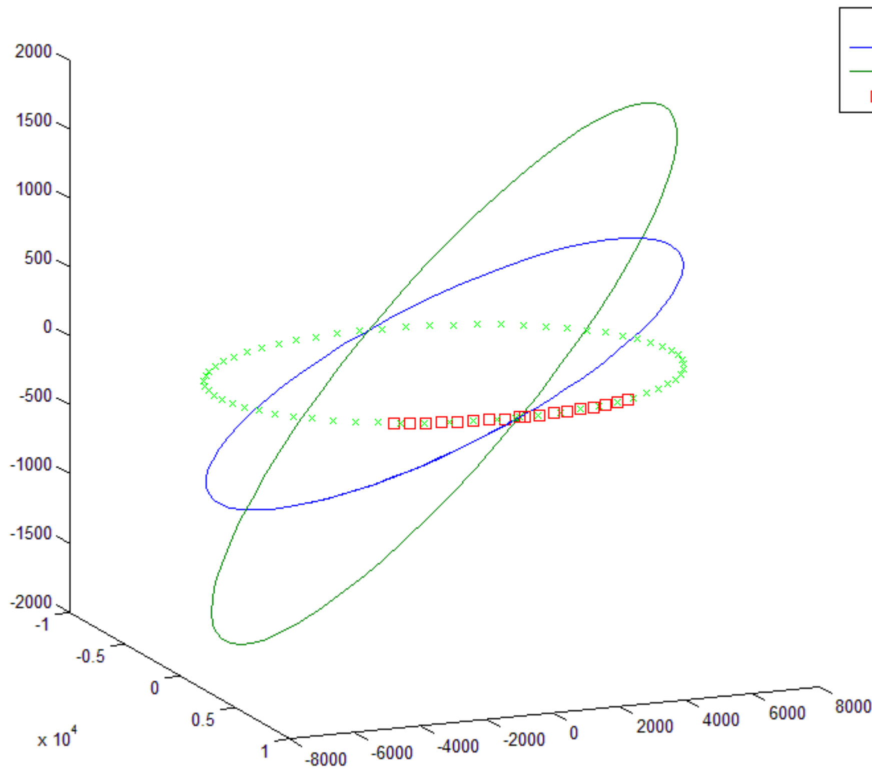

Figure 29: Mission 3 One Burn Centered at Crossover Point

Figure 29 shows the thrust location for a 15 min duration segment centered over the crossover point. This thrust profile was a slight modification on the thrust profile seen in Figure 28. Centering the thrust on the crossover point resulted in a more optimal thrust location with less propellant and a larger inclination angle achieved.

Table 4: Mission 3 One Burn Analysis Method

	Mission Req's	Electric System Instantaneous burn	Thrust centered on crossover pt	Thrust beginning at crossover pt
Plane Change (deg)	15	7.39	5.70	5.06
Propellant Mass (kg)	< 80	27.65	22.84	23.04
Trip Time (days)	< 90	90.00	90.00	90.00
Delta V (m/s)	N/A	981.63	798.8	806.1

As seen in Table 4 the largest plane change resulted from the instantaneous burn but this method was also the least accurate. Updating the spacecraft position and velocity vector continuously resulted in a slower transfer. Having the thrust location centered on the crossover point resulted in a more optimal transfer than having the thrust begin at the crossover point. The effective change in inclination resulting from thrusting at the crossover point produced the largest change in inclination. As the thrust location moved farther away from the crossover point the change in inclination resulting from thrusting was not as effective. None of the thrust profiles were able to achieve the 15 degree plane change in the time constraint.

Two Burn Profile

The spacecraft was able to achieve the plane change faster through centering the thrust segments on the crossover points. The thrust durations at the crossover points were varied. If the electric propulsion system was unable to achieve the 15 degree plane change the chemical propulsion system performed the remainder of the mission. Two analysis techniques were modeled, one which updated the position and velocity vector continuously and one that assumed smaller “instantaneous” changes in position and velocity. Figure 30 shows where the thrust was located for the “instantaneous” impulse cases.

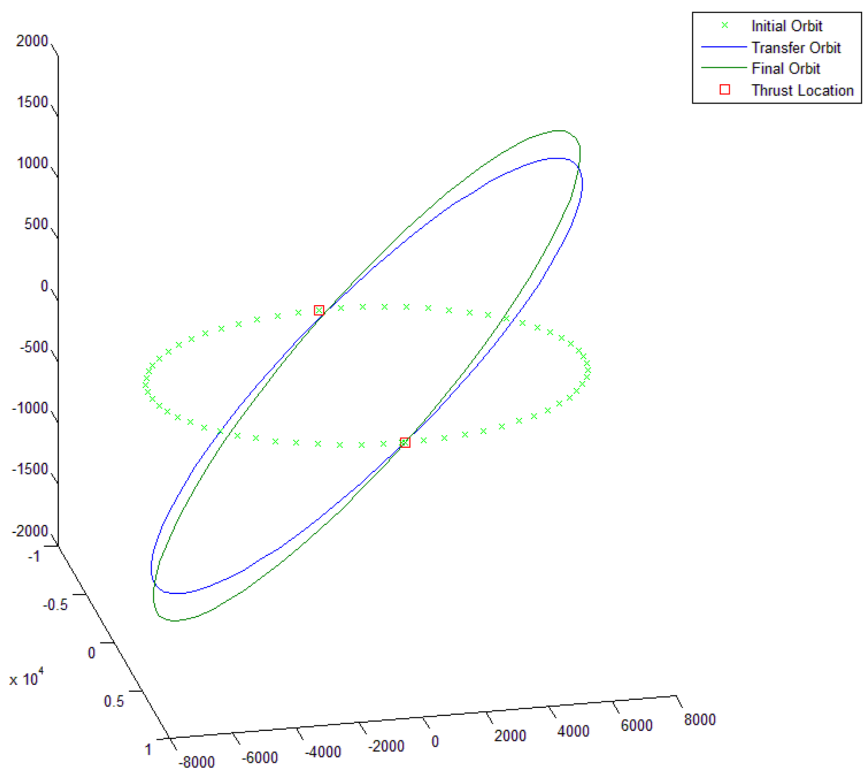

Figure 30: Mission 3 Electric Propulsion System Two Instantaneous Burns

The transfer orbit in Figure 30 with two 15 min burns achieved a much larger plane change than the single burn in Figure 27. The total change in the position and

velocity vector were applied at the thrust location. The transfer orbit, as seen in Figure 30, was 2 degrees short of the 15 degree plane change. The results for this thrust profile are shown in Table 5.

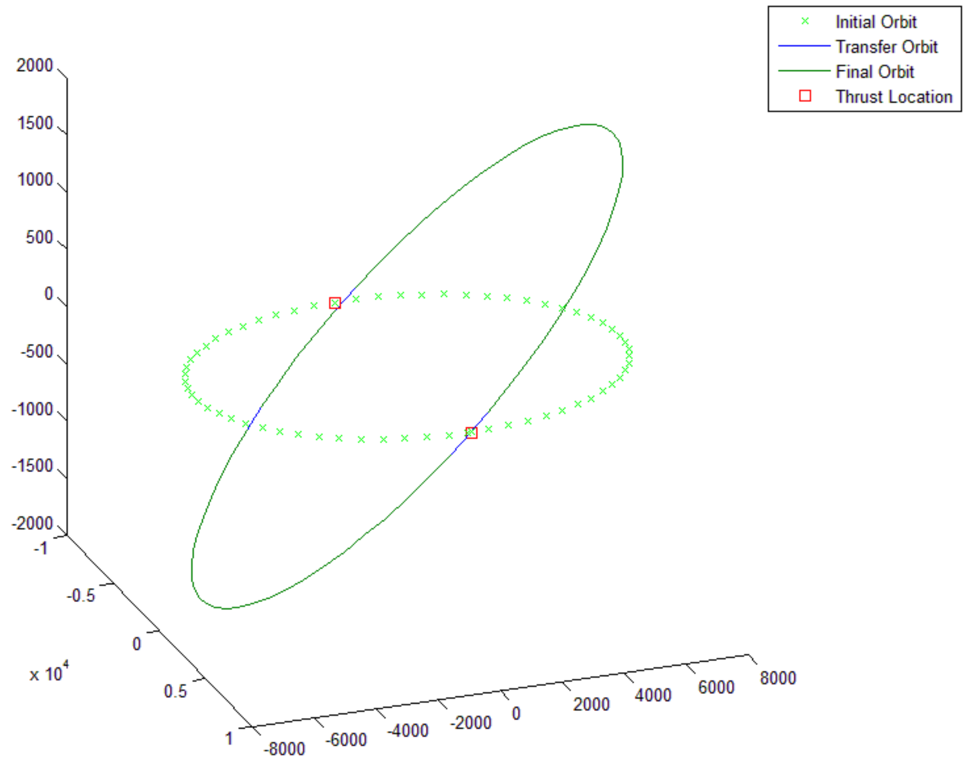

Figure 31: Mission 3 Combined System Orbit Profile

The combined system profile initially utilized electric thrust with two 15 min then utilized chemical thrust to complete the maneuver. This thrust profile was able to reach the final orbit in the 90 day time limit. This coupling made the final orbit achievable. The results for this thrust profile are shown in Table 5.

Table 5: Mission 3 Two Burn Analysis Method

	Mission Req's	Electric System 2 instantaneous burns	Thrust beginning at crossover pt	Thrust centered on crossover pt
Plane Change (deg)	15	13.01	10.93	12.43
Propellant Mass (kg)	< 80	46.08	46.07	45.97
Trip Time (days)	< 90	90.00	90.00	90.00
Delta V (m/s)	N/A	1740.71	1740.3	1735.9

As seen in Table 5 the instantaneous burn method over predicted the plane change amount and under predicted the required propellant. The final thrust profiles that were modeled are represented by Figure 32.

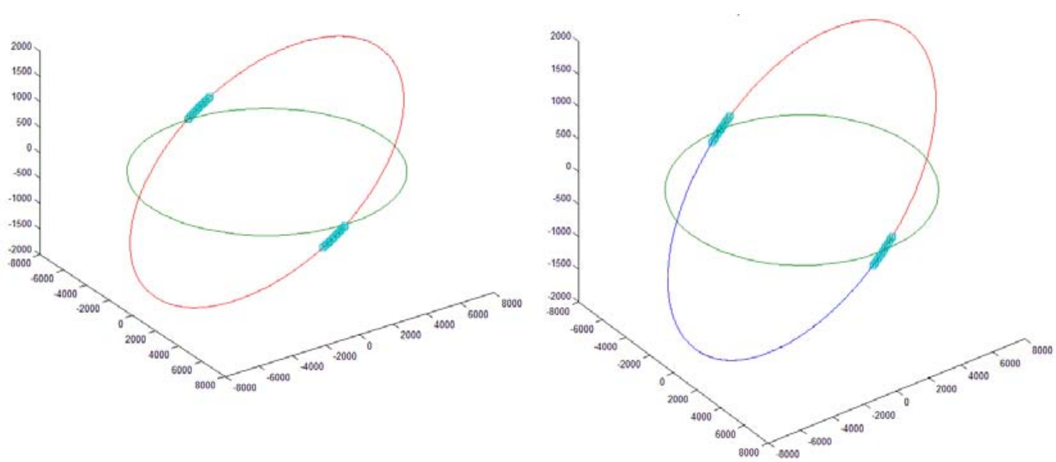

Figure 32: Mission 3 Two Burn Location Variation

The squares represent the area where the thrust occurred. The plot on the left of Figure 32 represents the case where the thrust began at the crossover point and the plot on the right represents the thrust centered over the crossover point. The thrust durations for the profile shown in Figure 32 were varied from 8 minutes at each crossover point to continuous thrusting. Thrust beginning at the crossover point was represented in the left hand plot in Figure 32 resulted in more propellant required to complete the plane change maneuver.

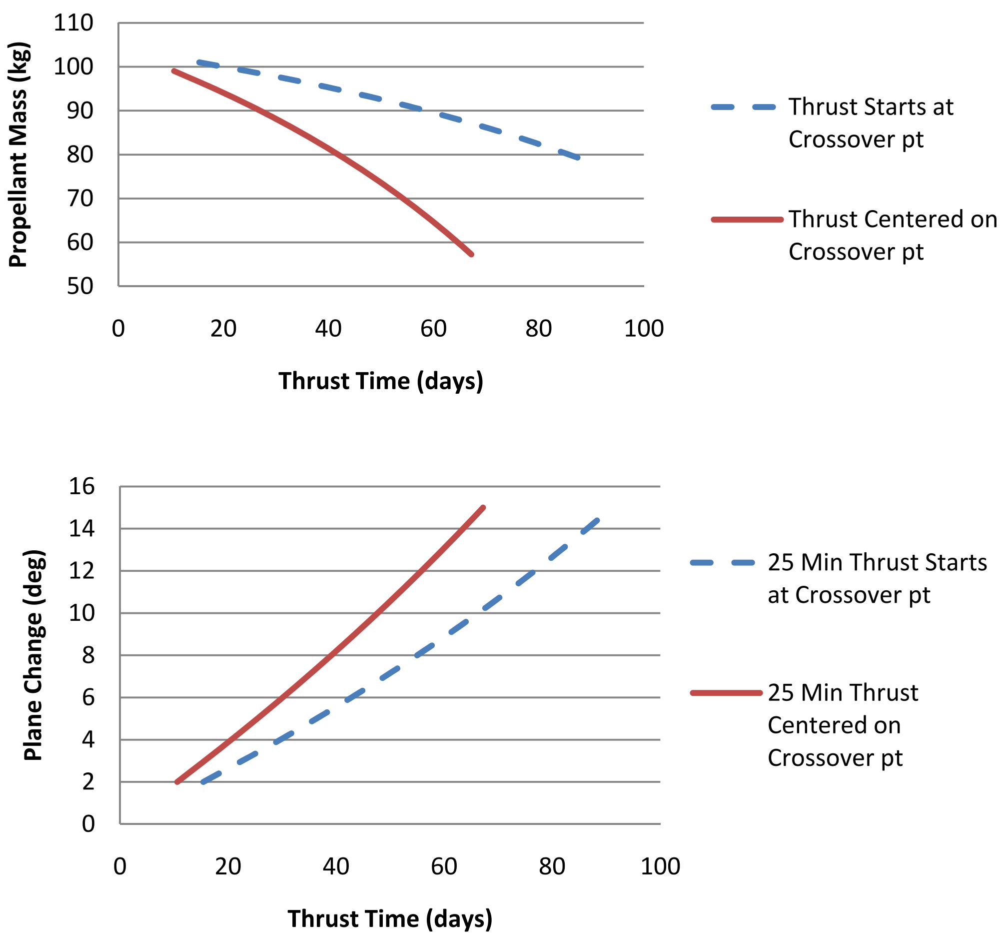

Figure 33: Mission 3 Effect of Thrust Location

As can be seen from the Figure 33 thrust beginning at the crossover point resulted in a longer transfer time and larger propellant required than when the thrust was centered

at the crossover point. The cause of this was because the farther away from the crossover point the less effect it has in changing the inclination. The greatest increase in inclination for a constant thrust resulted in the area immediately around the crossover point. That resulted in the thrust beginning at the crossover point requiring more time and propellant to complete the maneuver.

Thrust duration and required propellant had a direct correlation; the smaller the thrust duration the lower the propellant required. Increased thrust duration resulted in increased trip time. This relationship may be seen in Figure 34.

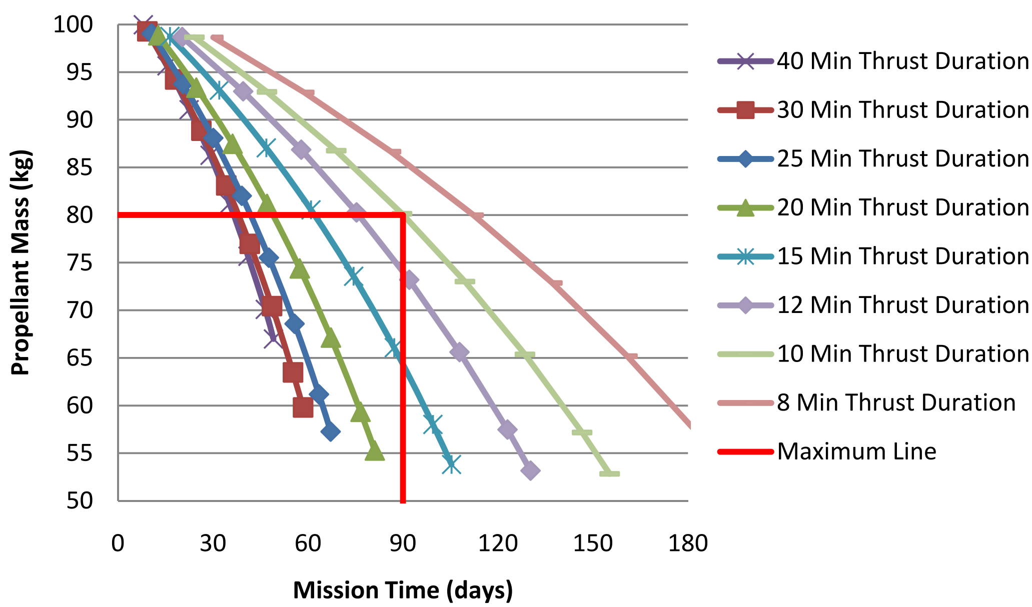

Figure 34: Mission 3 Combined System 15 Deg Plane Change

The red box in the lower corner of Figure 34 represents the portion of the different thrust duration profiles which would be able to achieve the 15 degree plane change in the time and propellant constraint. The point that results in the lowest propellant usage for each “thrust duration” line represents a maneuver completed entirely with electric propulsion. As the thrust duration decreased the mission time increased and

the propellant mass decreased. The required propellant decreased the most for the longer duration thrust profile. The time also decreased substantially with the increase in thrust duration per orbit. The propellant mass decreased slightly but after a time of about 110 days the propellant mass savings was minimal. The fastest transfer time for a 15 degree plane change was around 38 days which required just less than 80 kg of propellant.

Figure 35 displays the relationship of the thrust duration for a 15 degree plane change in a 90 day time constraint. The combined system shown in Figure 35 represents a system where the electric propulsion system utilized the majority of the mission time to perform the plane change maneuver and the chemical system performed the remainder of the plane change.

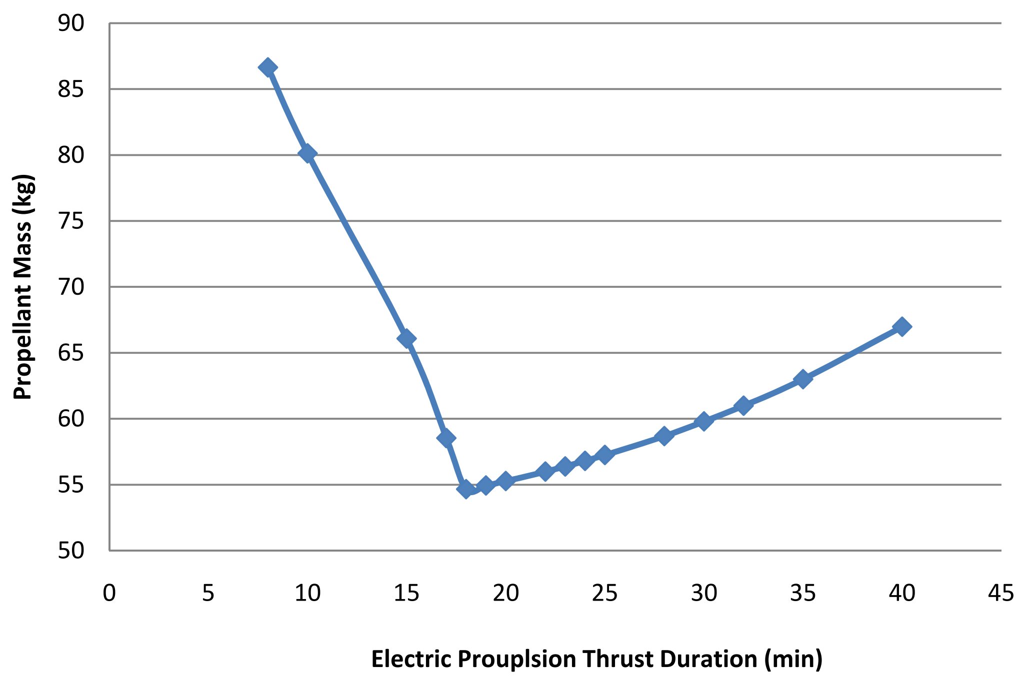

Figure 35: Mission 3 15 Deg Plane Change Trade Space

The minimum propellant required for a 15 degree plane change in the 90 day time constraint was achieved by thrusting centered at each crossover point for around 18 min.

Thrusting for 18 min resulted in 54.66 kg of propellant. The propellant mass required decreased as the thrust duration decreased until the 18 min point. This was caused by the plane change not being able to be completed entirely by the electric propulsion system. The assist of the chemical propulsion system caused the sharp increase in propellant required. Once the point was reached of the electric propulsion system being unable to perform the maneuver entirely on its own the shorter electric propulsion thrust duration resulted in the electric propulsion system performing larger portions of the plane change maneuver. If the maximum trip time were increased then the minimum propellant point in Figure 35 would continue to move down and to the left. Figure 36 displays this for a variety of thrust durations.

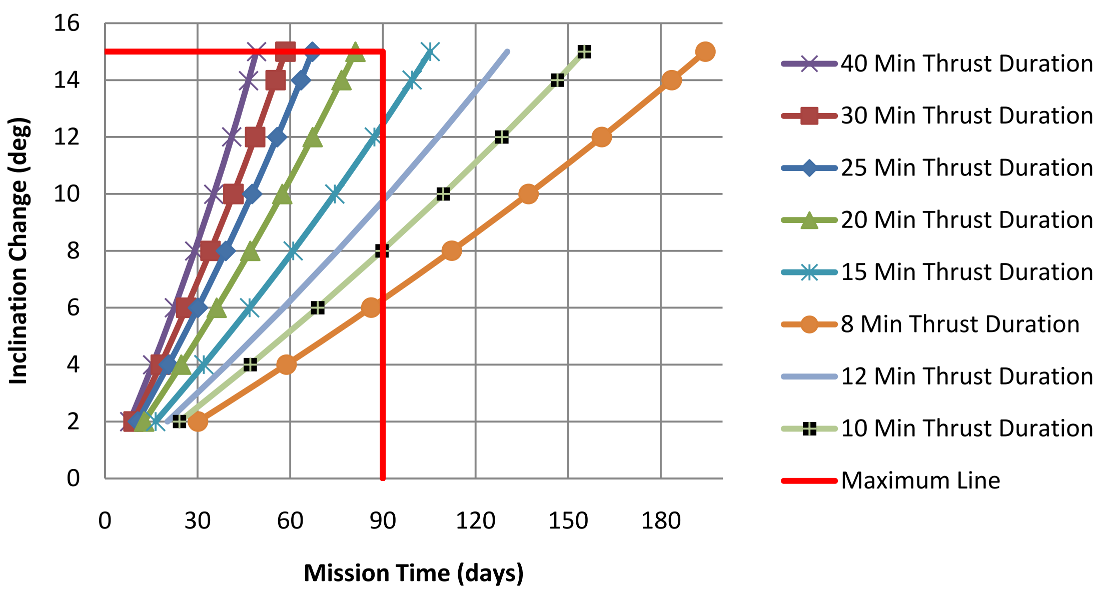

Figure 36: Mission 3 Inclination Change and Mission Time Relationship

As can be seen from Figure 36, the trip time greatly increased as the thrust duration decreased. The longer mission time corresponds to greater usage of electric propulsion; the longest mission time value corresponds to an entirely electric propulsion

maneuver. The variance in time was much smaller for smaller plane change angles. The difference in time ranges from around 22 days for a 2 degree inclination change to around 150 days for a 15 degree plane change. This plot corresponds with Figure 34.

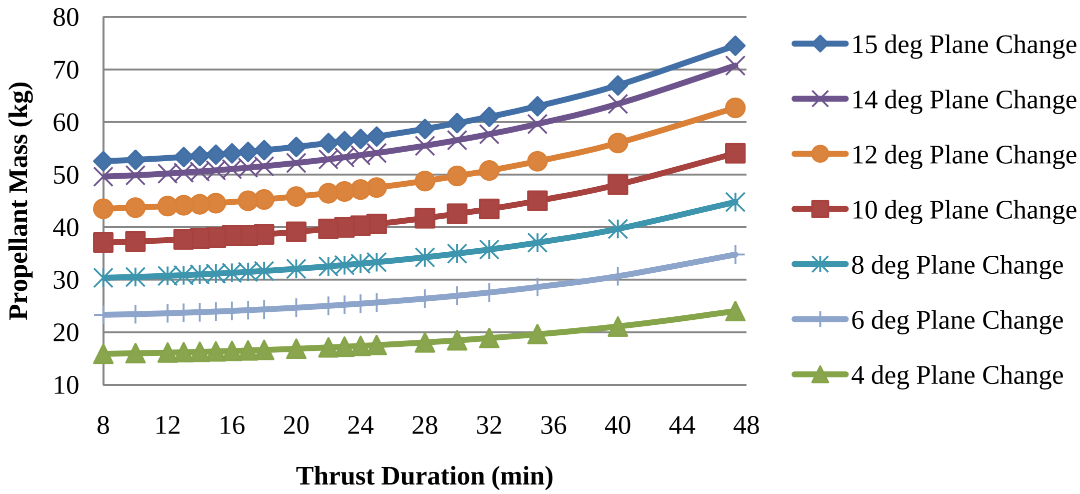

Figure 37: Mission 3 Effect of Thrust Duration on Propellant Mass

As seen from Figure 37 the curves representing the propellant mass with respect to thrust duration were very uniform. The spacing between the lines decreased slightly as the plane change increased. For each of the plane change amounts the propellant mass savings through longer thrust durations had a smaller payoff. The slope of the propellant verses thrust time line flattens out until the mass savings is less than around 0.1 kg. Each plane change amount resulted in similar trends for the change in propellant with respect to the thrust duration. The 15 degree plane change is represented in the following figures.

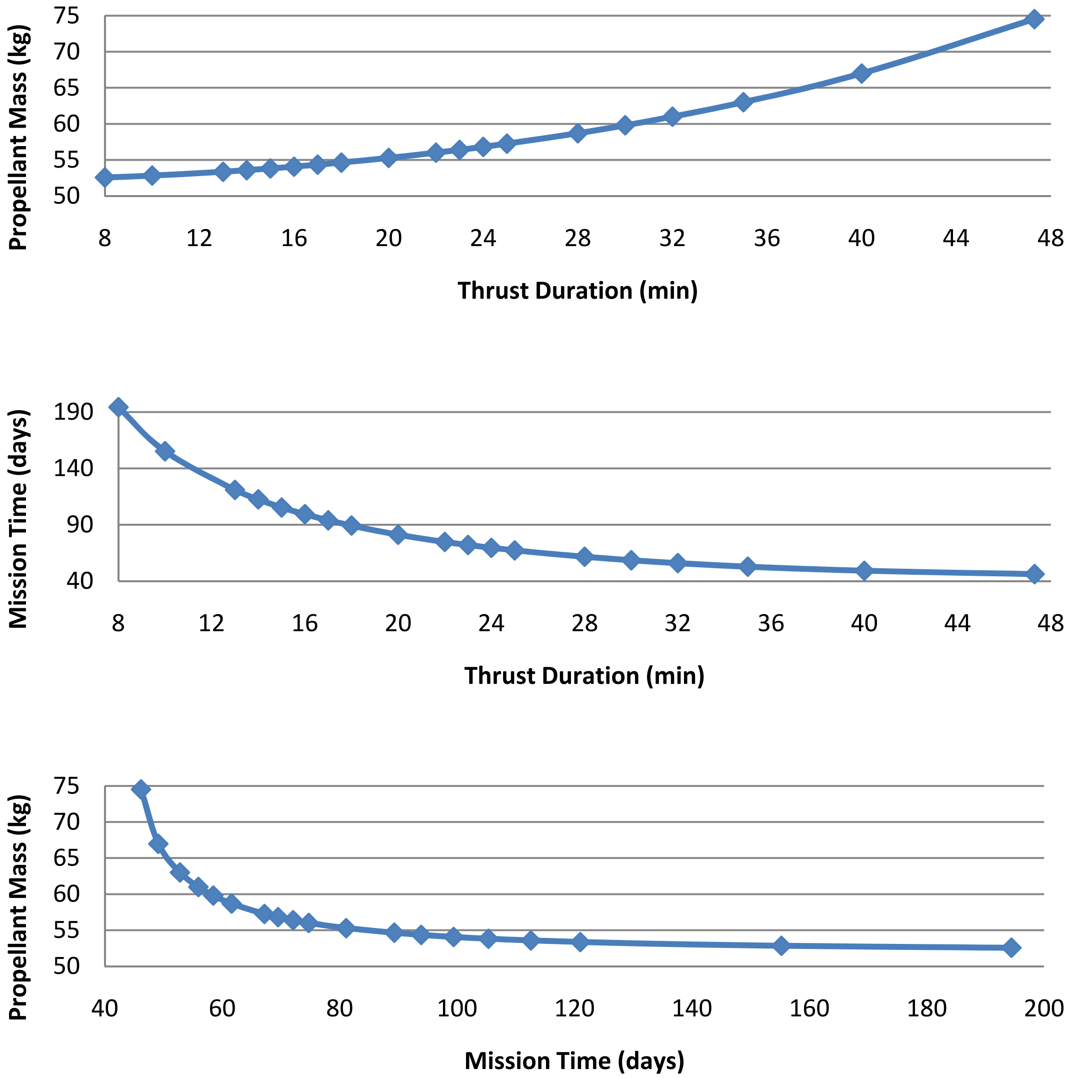

Figure 38: Mission 3 Two Burn 15 deg Plane Change

Figure 38 displays the values for a purely electric propulsion plane change maneuver. The least propellant resulted from the shortest thrust duration as seen in the uppermost plot. That same thrust duration would take 190 days to complete the 15 degree maneuver. The final data point on the upper two plots represents the thrusting continuously case. That point corresponded to the highest required propellant at around 75 kg.

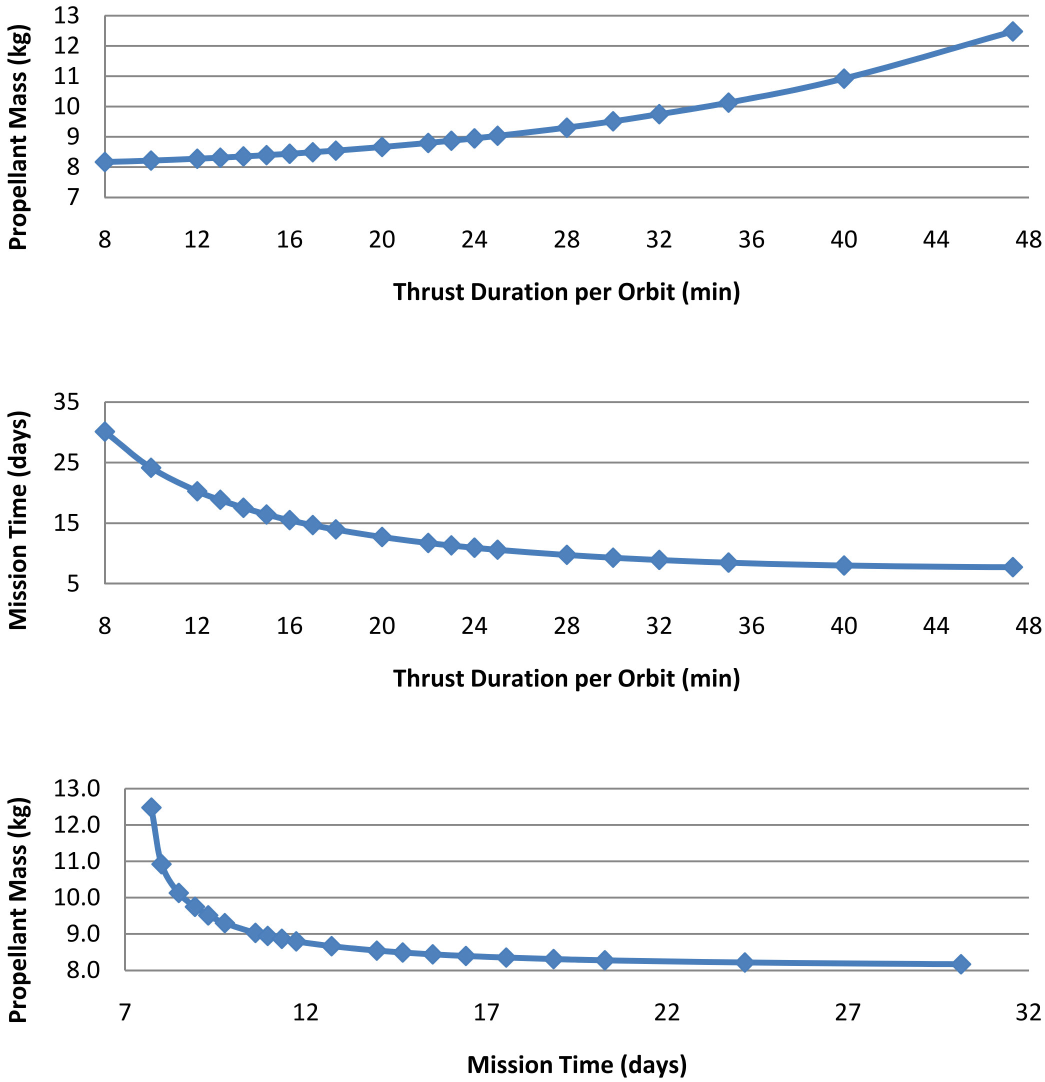

Figure 39: Mission 3 Two Burn 2 deg Plane Change

The trends seen in Figure 39 for a 2 degree plane change are very similar to those viewed in Figure 38 for a 15 degree plane change. Thrusting continuously would achieve a 2 degree plane change in around 30 days with about 8 kg of propellant. A pure chemical 2 degree plane change would require 19 kg of propellant.

Figure 40 represents a thrust profile where continuous thrusting occurred.

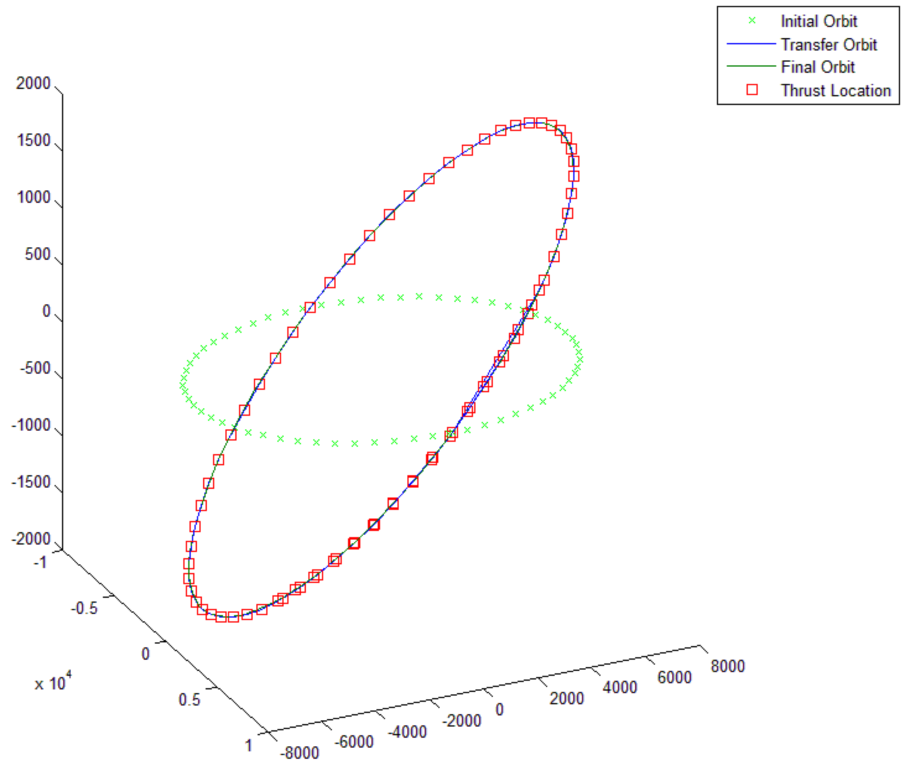

Figure 40: Mission 3 Continuous Thrusting

The continuous thrusting case shown in Figure 40 achieved the final orbit with the least amount of time for the electric propulsion profiles. All the values in Table 6 are for the transfer where the position and velocity were continuously updated based on the thrust duration. The values listed match up with those shown in Figure 34 - Figure 38.

Table 6: Mission 3 15 deg Plane Change Values

	Mission Req's	Cont. Thrust	40 Min Thrust Duration	30 Min Thrust Duration	20 Min Thrust Duration	18 Min Thrust Duration	15 Min Thrust Duration
Propellant Mass (kg)	< 80	74.53	66.98	59.80	55.28	54.94	66.08
Trip Time (days)	< 90	46.16	49.11	58.47	81.11	89.32	87.25
Delta V (m/s)	N/A	3146.6	2739.4	2376.9	2159.3	2128.3	2073.3

The thrust duration that resulted in the least required propellant was an 18 min thrust duration centered at each crossover point. The propellant required decreased as the thrust duration decreased. The 15 min thrust duration required more propellant than the 18 min thrust duration because the electric system alone was unable to achieve the plane change within the time constraint. The utilization of the chemical propulsion system to achieve the plane change within the time constraint caused the required propellant to increase.

Using the idealized impulsive instantaneous equations[2-5] resulted in the following:

Table 7: Textbook vs Detailed Analysis Values

	Textbook Plane Change	Best Detailed Analysis
Propellant Mass (kg)	51.52	54.94
Trip Time (days)	N/A	89.32
Delta V (m/s)	1987.31	2128.3

The textbook equations resulted in an underestimate of the required delta-v for the electric propulsion system which is reflected in the decreased required propellant mass. The impulsive equation also does not give any estimate of total trip time which is very substantial when utilizing an electric propulsion system.

Mission Summary

The primary conclusions from each of the missions are re-iterated below. The use of electric propulsion in all the missions modeled resulted in a decrease in propellant mass. The utilization of an electric propulsion system had varying effects for the different missions. The use of an electric propulsion system had potentially the largest impact on a plane change maneuver.

Mission 1: 180 deg Phase Change in 12 hr

Utilizing electric propulsion for a portion of the mission time constraint would result in mass savings. Longer duration time constraints results in additional mass savings. Varying the thrust profile decreased the necessary propellant mass slightly, but

the largest driver for propellant mass required in a large rephase mission was the time constraint. A large time constraint would also result in a reduction in mass for the chemical propulsion system. The chemical propulsion system required propellant mass for rephase maneuvers with long time constraints resulted in the chemical propulsion system propellant approaching the electric propulsion required mass for that same time constraint. Even though the chemical propulsion system required propellant approached the electric propulsion required propellant electric propulsion always required around 1 kg less propellant to complete the maneuver.

Mission 2: 1000 km Altitude Increase in 12 hours and return in 48 hours

Utilizing a multi-mode propulsion system decreased the propellant mass required from 35 kg to 30 kg for a 1000 km altitude change maneuver with a 48 hr time constraint. The entire mission with an altitude raise and a return in 30 days would require around 70 kg of propellant for a chemical propulsion system to perform alone. With a combined system the total propellant used would be reduced to 44.85 kg which is a decrease of 36% in propellant use. The best thrust profile was to initially use electric propulsion followed by the chemical propulsion system performing a small Hohmann maneuver to increase the altitude with the final altitude increase being made by the electric propulsion system. This thrust profile resulted in 30.24 kg of propellant required to perform the maneuver within the 12 hour time constraint. The first electric propulsion segment used 1.58 kg of propellant, the chemical propulsion segment used 27.07 kg propellant, and the second electric propulsion segment used 1.58 kg of propellant. The chemical – electric profile resulted in 30.25 kg of required propellant, with the electric propulsion system using 3.16

kg propellant and the chemical propulsion system using 27.08 kg propellant. In general, thrust profiles that utilized chemical followed by electric propulsion resulted in the largest propellant savings.

Mission 3: 15 deg Plane Change in 90 days

For the plane change mission using simplifying assumptions resulted in an under prediction of the trip time and propellant mass required for the maneuver. The location of the thrust for the electric propulsion segment had a large effect on the amount of propellant required. The larger the trip time constraint the lower the propellant required to perform the maneuver. Using the same equations for a chemical plane change as an electric propulsion plane change under-predicts the delta-v and the required propellant and does not give a way to estimate the trip time. The fastest the MMP system would be able to perform the mission within the mass constraint was in 37 days. The electric propulsion system required half the propellant of the chemical propulsion system to perform the plane change. Plane change missions are very costly, but using optimal thrust profiles results in a drastic reduction in propellant required.

Conclusion

The full potential of electric propulsion will not be realized until it begins to be modeled in conjunction with chemical propulsion systems. When electric propulsion systems are used with chemical systems for missions, as seen in missions described above, a mass savings results. Treating them separately backs the electric propulsion system into a corner through limiting it only to station-keeping missions or missions with

very long time constraints. Also, electric propulsion systems can be utilized for plane change missions.

The mass savings resulting from a multi-mode propulsion system was not analyzed herein but there has been a documented mass savings from having an integrated electric and chemical propulsion system. The utilization of an integrated system promotes coupled utilization. Numerous papers have been written regarding electric propulsion development and missions in which electric propulsion are generally used have been documented. Improvements in multi-mode propulsion systems will hopefully produce a new grouping of papers addressing improved mission performance through the use of a combined system.

Future Work

For this study the spacecraft mass was broken into two segments, the available propellant (80 kg), and the remainder of the spacecraft (100 kg). The overall spacecraft could be modeled in more detail. The comparison of the chemical propulsion only system would have slightly different results if the propulsion systems were sized. The propellant mass could be kept at 80 kg but the difference resulting from a chemical, electric, or multi-mode propulsion system would be reflected in the payload mass of the system.

The chemical and electric propulsion systems were treated as systems that had a set specific impulse and thrust level. The electric propulsion code was written to have available power as input but varying power levels were not analyzed. An assumption made was that the power level would remain constant during the transfer, but it would not

be that difficult to incorporate the effect of thrust into the electric propulsion calculations. The MATLAB scripts were written to include a power variable but it was not included in calculations. The efficiencies of different thrusters could also be included. The codes are set up to calculate the separate missions. Incorporating additional spacecraft information would be relatively simple.

Drag calculations were not included in the delta-v budget, but for lower orbits drag could become a driving factor. Including a calculation of what the drag would be on the spacecraft could be incorporated into the necessary delta-v budget to complete maneuvers. If power was varying the electric propulsion system may not have a constant thrust.

A separate mission that was proposed but was not analyzed was an orbit lowering and drag make-up mission. This mission could utilize the orbit raising and lower code developed for Mission 1, but would add the increased difficulty of modeling the effect of drag on a spacecraft at lower altitudes. The largest problem that would occur with modeling a LEO drag makeup mission would be atmospheric density to use. This value is not known and fluctuates greatly with the sun solar cycle. Atmospheric density can fluctuate orders of magnitude causing the required delta-v to compete with the increased drag to change by orders of magnitude.

Bibliography

[1]Petter, M., Holmes, M.R.., "Multiple Mode Propulsion system for Spacecraft Orbit Transfer, Maneuvering, and Attitude Control," BAA-06-02-PKTB. (24 April 2006).

[2]Bate, R., Mueller, D., & White, J., (1971). *Fundamentals of Astrodynamics*. Dover Pubns.

[3]Curtis, H., *Orbital Mechanics for Engineering Students.* Elsevier Ltd., 2005.

[4]Humble, R., Henry, G., & Larson, W. (1995). *Space propulsion analysis and design*. Learning Solutions.

[5]Larson, W., and Wertz, J., *Space Mission Analysis and Design.* Space Technology Library. (El Segundo, CA) 1999.

[6]Stuhlinger, E., "Origin and Early Development of Electric Propulsion Concepts", AIAA Paper 91-3442, September 1991.

[7]Edelbaum, T.N., "The use of high- and low-thrust propulsion in combination for space missions," Journal of the Astronautical Sciences, Vol. 9, No 2, 1962, pp. 49-69

[8]Martinez-Sanchez, M., and Pollard, J. E., "Spacecraft Electric Propulsion: An Overview," *Journal of Propulsion and Power*, Vol. 14, No. 5, 1998, pp. 688-699.

[9]Cassady, R., King, D., and Meckel, N, "Repositioning Mission Benefits Comparison for Near-Term Electric Propulsion Technology," 30th AIAA/ASME/SEA/ASEE Joint Propulsion Conference (Indianapolis, IN), AIAA 94-3002.

[10]Pidgeon, D., Corey, R., and Sauer, B., "Two Years On-Orbit Performance of SPT-100 Electric Propulsion," 24th AIAA International Communications Satellite Systems conference (San Diego, CA). AIAA 2006-5353.

[11]Grys, K., Rayburn, C., Wilson, R., Fisher, J., "Multi-Mode 4.5 KW BPT-4000 Hall Thruster Qualification Status," 39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference. (Huntsville, AL). AIAA 2003-4552.

[12]Jones, J.M., Einhorn, R.S., "The USAF Electric Propulsion Systems Activities, 1993" 29th AIAA/SAE/ASME/ASEE Joint Propulsion conference. (Monterey, CA). AIAA 93-1937.

[13]Tverdokhlebov, S.O., et. al, "Overview of Electric Propulsion Activities in Russia," 40th AIAA/SAE/ASME/ASEE Joint Propulsion Conference. (Fort Lauderdale, FL). AIAA 2004-3330.

[14]Koppel, C., Marchandise, F., Estublier, D., Jolivet, J.,"The SMART-1 Electric Propulsion Subsystem In Flight Experience," 40th AIAA/SAE/ASME/ASEE Joint Propulsion Conference. (Fort Lauderdale, FL). AIAA 2004-3435.

[15]Janson, S., "The On-Orbit Role of Electric Propulsion," 29th AIAA/SAE/ASME/ASEE Joint Propulsion Conference. (Monterey, CA). AIAA 93-2200.

[16]Free, B., and Babaska, V., "Reposition of Geostationary Spacecraft: Chemical and Electric Propulsion Options," AIAA International Communications Satellite Systems Conference, (Washington D.C.),1996, pp. 1302-1311.

[17]Kluever, C., "Spacecraft Optimization with Combined Chemical-Electric Propulsion," *Journal of Spacecraft and Rockets*, Vol. 32, No. 2, 1994, pp. 378-380.

[18]King, S., Walker, M, Kluever, C., "Small Satellite LEO Maneuvers with Low-Power Electric Propulsion," 44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, AIAA 2008-4516.

[19]Mailhe, L., and Heister, S., "Design of a Hybrid Chemical/Electric Propulsion Orbital Transfer Vehicle," *Journal of Spacecraft and Rockets*, Vol. *39*, No. 1, 2002, pp. 131-139.
[20]Ulybyshev, Y., "Optimization of Multi-Mode Rendezvous Trajectories with Constraints," Cosmic Research, 2008. Vol. 46., No 2., pp. 133 - 145
[21]Zakirov, V., Richardson, G., Sweeting, M., "Surrey Research Update on N2O Catalytic Decomposition for Space Applications," 37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference (Salt Lake City, UT) AIAA 2001-3922

Appendix A: Mission 1 MATLAB Code

Five separate MATLAB scripts were written to model the phase change mission. The main file is Run_M1.m, it is the master file and any changes in the vehicle information will be utilized in the other codes. The primary scripts which calculate the transfers are chem._rephase.m and ep_rephase.m. The rephase.m file utilizes chem._rephase.m and ep_rephase.m to calculate the pure chemical and electric transfer orbits. The chem._rephase_ts.m file and the ep_rephase_time.m file calculates the combined system values. The order the scripts are listed in this file are:

1) Run_m1.m

2) rephase.m

3) chem_rephase.m

4) ep_rephase.m

5) chem_rephase_ts.m

6) ep_rephase_time.m

```
%This script runs Mission 1: 12 hr 180 deg rephase, with a 12 hr
return.
%This program calls rephase.m, which calls chem_rephase.m and
ep_manuever.m
%Written by Tiffany Rexius 2010-2011
clc;
clear;
%***************Vehicle Information************************
m_init      = 180;  %[kg]
P_avail     = 1000; %[W]
Isp_ep      = 600;  %[s]
Thrust_ep   = 0.11; %[N]
Isp_chem    = 235;  %[s]
Thrust_chem = 18;   %[N]
%**********************************************************
%********************Constants****************************
mu      = 3.986e5;  %[km^3/s^2]
g0      = 9.81;     %[m/s^2]
r_earth = 6378;     %[km]
hr2sec  = 60*60;    %[sec/hr]
sec2hr  = 1/hr2sec; %[hr/sec]
```

```
deg2rad = pi/180;   %[rad/deg]
rad2deg = 180/pi;   %[deg/rad]
km2m    = 1000;     %[m/km]
m2km    = 1/1000;   %[km/m]
%*************************************************************
%*********************Mission 1******************************
r_init      = 500 + r_earth; %[km]
t_const     = 12*hr2sec;     %[sec] Time constraint
rephase_ang = 180;           %[deg] Rephase Angle
imp         = 20*60;         %[sec] Length of burn for chemical

[nu_diff,t_ch,dv_ch,mp_ch,t_ep,dv_ep,mp_ep,Th_ch] =
rephase(r_init,rephase_ang,Isp_ep,Thrust_ep,Isp_chem,t_const,m_init,P_a
vail,imp);

%The average thrust for electric
Th_ep = m_init*dv_ep(end)/(t_ep(end)*hr2sec); %[N]
fprintf('******Results for Mission 1********\n\n')
fprintf('   Rephase Angle: %6.2f\n\n',nu_diff)
fprintf('**********Chemical System**********\n')
fprintf('   Transfer time per Man. (hr) = %6.2f\n',t_ch)
fprintf('   Delta V (m/s)               = %6.2f\n',dv_ch*2)
fprintf('   Propellant Mass (kg)        = %6.3f\n',mp_ch*2)
fprintf('   Average Thrust (N)          = %6.2f\n\n',Th_ch)
fprintf('*************EP System*************\n')
fprintf('   Transfer time per Man. (hr) = %6.2f\n',t_ep)
fprintf('   Delta V (m/s)               = %6.2f\n',dv_ep(end)*2)
fprintf('   Propellant Mass (kg)        = %6.2f\n',(m_init - mp_ep)*2)
fprintf('   Average Thrust (N)          = %6.2f\n\n',Th_ep)

%This Runs a sweep of various rephase angles
rephase_ts(r_init,Isp_ep,Thrust_ep,Isp_chem,t_const,m_init,imp)
chem_rephase_ts;
```

---

rephase.m file

```
function [nu_diff, t_ch,dv_ch,mp_ch,ti_ep,dv_ep,mp_ep,Th_ch] =
rephase(r_init,rephase_ang,Isp_ep,Thrust_ep,Isp_chem,time_sec,m_init,P_
avail,imp)

%Written by Tiffany Rexius 2010 - 2011
%Calculates transfer time, delta v, and propellant mass for an EP
system as
%well as for a chemical system.

%For the chemical system the max. time constraint is enforced, it is
not
%enforced for the ep system.

%8/17/10 T. Rexius Models the ep and chem systems seperatly, plots
orbit trace
%with markings of where each spacecraft would be.  Works for any input
%rephase angle.
```

```
format long

%********************Constants***************************
mu      = 3.986e5;  %[km^3/s^2]
g0      = 9.81;     %[m/s^2]
r_earth = 6378;     %[km]
hr2sec  = 60*60;    %[sec/hr]
sec2hr  = 1/hr2sec; %[hr/sec]
deg2rad = pi/180;   %[rad/deg]
rad2deg = 180/pi;   %[deg/rad]
km2m    = 1000;     %[m/km]
m2km    = 1/1000;   %[km/m]
%******************************************************
%*******************Convert Units*************************
r_ang   = rephase_ang*deg2rad;       %[rad]
t_max   = time_sec;                  %[sec]
t_tot   = t_max;                     %[sec]

a_init     = r_init;                 %[km]Circular orbit
P_init     = 2*pi*sqrt(a_init^3/mu); %[sec]
omega_init = (2*pi)/P_init;          %[rad/sec]
%******************************************************
%***************Chemical Propulsion Only******************
%Calculates a transfer into a "hohmann" type orbit for a certain number
of
%orbits until it reenters its original orbit at a phase change angle
equal
%to the input angle and in just under the necessary time allotment.

nu_init = 0; %Initial position in orbit
[pos1,pos2,vel_chem,t_chem,mp_chem,dv_chem,nu_diff,Th_ch] =
chem_rephase(m_init,r_init,Isp_chem,r_ang,t_max,nu_init,imp);

r_chem_x = pos1(1,:);  %[km]
r_chem_y = pos1(2,:);  %[km]

%***************Electric Propulsion Only******************
%Only Electric propulsion transfer.  This disregards the trip time
%constraint and calculates the amount of time it would take for the
%rephase angle if it was thrusting constantly.

t_max_a      = t_max*6.0;  %[sec] Removes time constraint for this case
rephase_diff = 0;        %[rad] Initial rephase difference
check        = 1;        %1 = decrease radius, 2 = increase radius

[rephase_tot,nu,pos,vel,t_ep,mp,r_new,dv_tot,Th_ep] =
ep_rephase(r_init,m_init,t_max_a,r_ang(end),rephase_diff,Isp_ep,Thrust_
ep,check);

rephase_deg = rephase_tot(end)*rad2deg;
r_ep_x      = pos(1,:);  %[km]
r_ep_y      = pos(2,:);  %[km]
```

```
%********************Set up Plots************************
%Plot outline of earth on plot and outline of initial orbit
nu_earth = 0;          %[rad]
nu_step  = 1*deg2rad;  %[rad]
kk       = 1;

while nu_earth < 2*pi
    r_earth_x(kk) = r_earth*cos(nu_earth);
    r_earth_y(kk) = r_earth*sin(nu_earth);
    rix(kk)       = r_init*cos(nu_earth);
    riy(kk)       = r_init*sin(nu_earth);
    nu_earth      = nu_earth + nu_step;
    kk            = kk+1;
end

rixe = r_init*cos(omega_init*t_ep(end));
riye = r_init*sin(omega_init*t_ep(end));

figure(1)
subplot(2,1,1)
plot(r_ep_x,r_ep_y,'--g',rix,riy,'-k',r_earth_x,r_earth_y,'-
b',r_ep_x(end),r_ep_y(end),'x',rixe,riye,'o')
title('Electric Propulsion Transfer Orbit')
legend('Transfer Orbit [km]','Initial Orbit [km]','Earth [km]')
subplot(2,1,2)
plot(r_chem_x,r_chem_y,'--g',rix,riy,'-k',r_earth_x,r_earth_y,'-
b',r_chem_x(end),r_chem_y(end),'x',pos2(1,:),pos2(2,:),'o')
title('Chemical Propulsion Transfer Orbit')
legend('Transfer Orbit [km]','Initial Orbit [km]','Earth
[km]','Transfer Orbit Position', 'Initial Orbit Position')

figure(2)
plot(t_ep*sec2hr,rephase_tot*rad2deg)
title('Rephase Time vs. Rephase Degree for EP system')
xlabel('Time (hr)')
ylabel('Rephase Amount (deg)')

t_ch   = t_chem(end)/3600;
dv_ch  = dv_chem(end)*km2m;
mp_ch  = mp_chem(end);
ti_ep  = t_ep(end)/3600;
dv_ep  = dv_tot(end);
mp_ep  = mp;
```

Chem._rephase.m file

```
function [pos1,pos2,vel,time,mp,dv,nu_diff,Th_ch] =
chem_rephase(m_init, r_init,Isp,r_ang,t_tot,nu_init,imp)

%Written by Tiffany Rexius 2010 - 2011

format long
```

```
%*********************Constants*****************************
mu      = 3.986e5;  %[km^3/s^2]
g0      = 9.81;     %[m/s^2]
r_earth = 6378;     %[km]
hr2sec  = 60*60;    %[sec/hr]
sec2hr  = 1/hr2sec; %[hr/sec]
deg2rad = pi/180;   %[rad/deg]
rad2deg = 180/pi;   %[deg/rad]
km2m    = 1000;     %[m/km]
m2km    = 1/1000;   %[km/m]
%***********************************************************

r_min   = r_earth + 300;           %[km]
a_init  = r_init;                  %[km] Circular orbit
P_init  = 2*pi*sqrt(a_init^3/mu);  %[sec]
v_init  = sqrt(mu/r_init);         %[km/s]
w_init  = (2*pi)/P_init;           %[rad/s]

a_trans  = (mu*(w_init + r_ang/t_tot)^(-2))^(1/3); %[km]
r_trans  = a_trans*2 - r_init;                     %[km]

%Finds Optimal Transfer Orbit based on time constraint
flag = 1;
k    = round(t_tot/hr2sec);
while flag == 1
    for i = 1:k
        p_trans(i) = P_init*(r_ang/(2*pi*i) + 1);
        w_trans(i) = (2*pi)/p_trans(i);
        A_trans(i) = (mu*(p_trans(i)/(2*pi))^2)^(1/3);
        r_trans(i) = A_trans(i)*2 - r_init;
        v_apo(i)   = sqrt(mu*(2/r_init-1/A_trans(i)));
        dv_inst(i) = abs(v_init - v_apo(i));
        dv_tot(i)  = 2*dv_inst(i)*km2m;
        m_prop(i)  = m_init - m_init*exp(-dv_tot(i)/Isp/g0);
        nu_diff    = abs(w_trans(i)*p_trans(i)*i -
w_init*p_trans(i)*i)*rad2deg;
        nu_sat     = w_init*p_trans(i)*i;
        sat_initx  = r_init*cos(w_init*p_trans(i)*i);
        sat_inity  = r_init*sin(w_init*p_trans(i)*i);
    end
    if p_trans(i)*i > t_tot
        k = k - 1;
        p_trans = 0*ones(1,k);
        w_trans = 0*ones(1,k);
        A_trans = 0*ones(1,k);
        r_trans = 0*ones(1,k);
        v_apo   = 0*ones(1,k);
        dv_inst = 0*ones(1,k);
        dv_tot  = 0*ones(1,k);
        m_prop  = 0*ones(1,k);
    else
        flag    = 2;
        dv_chem = dv_tot(i)*m2km;
```

```
    end
end

P_trans      = p_trans(end);        %[sec]
a_trans      = A_trans(end);        %[km]
time         = i*P_trans;           %[sec]
Energy_tr    = -mu/(2*a_trans);     %[km^2/s^2]
dv           = dv_chem;             %[km/s]
ecc          = 1 - r_init/a_trans;
per          = a_trans*(1-ecc^2);   %[km]
kkmax        = 2000;
t_step       = time/kkmax;          %[sec]
t_count      = 0;                   %[sec]

for kk = 1:kkmax
    r_trans      = per/(1 + ecc*cos(w_trans(end)*t_count + nu_init));
    v_trans(kk)  = sqrt(2*(Energy_tr+mu/r_trans(end)));
    r_x(kk)      = r_trans(end)*cos(w_trans(end)*t_count + nu_init);
    r_y(kk)      = r_trans(end)*sin(w_trans(end)*t_count + nu_init);
    t_count      = t_count + t_step;
end

mp          = m_init - m_init*exp(-dv*km2m/Isp/g0); %[kg]
pos1        = [r_x; r_y];                           %[km]
pos2        = [sat_initx; sat_inity];               %[km]
vel         = v_trans;                              %[km/s]
burn_time   = 2*imp;                           %[sec] for Hohmann 2
burns
Th_ch       = m_init*dv(end)*km2m/(burn_time); %[N]
```

---

ep._rephase.m file

```
function [rephase_tot, nu, pos, vel, time, mf, r_new, dv_tot, Th_ep] =
ep_rephase(r_init, m_init, t_max,r_ang, rephase_diff, Isp_ep,
Thrust_ep,check)

%Written by Tiffany Rexius 2010 - 2011
%Thrusts up until time constraint (t_max) is reached.

%*********************Constants*****************************
mu      = 3.986e5;  %[km^3/s^2]
g0      = 9.81;     %[m/s^2]
r_earth = 6378;     %[km]
hr2sec  = 60*60;    %[sec/hr]
sec2hr  = 1/hr2sec; %[hr/sec]
deg2rad = pi/180;   %[rad/deg]
rad2deg = 180/pi;   %[deg/rad]
km2m    = 1000;     %[m/km]
m2km    = 1/1000;   %[km/m]
%***********************************************************
Isp         = Isp_ep;                     %[sec]
m_dot       = Thrust_ep/(Isp*g0);         %[kg/s]
a_init      = r_init;                     %[km] if circular orbit
P_init      = 2*pi*sqrt(a_init^3/mu);     %[sec]
omega_init  = 2*pi/P_init;                %[rad/s]
```

```
Vel_init   = sqrt(mu/r_init);            %[km/s]

r_min        = 290 + r_earth;            %[km]
t_step       = 60;                       %[sec]
t_init       = 0;                        %[sec]
delta_v_tot  = 0;                        %[km/s]
prop_mass    = 0;                        %[kg]
nu           = 0;                        %[rad]
jj           = 1;                        %[]    Count
Vel_new      = Vel_init;                 %[km/s]
r_new        = r_init;                   %[km]
nu_old       = 0;                        %[rad]
tol          = 0.001;                    %[]    Tolerance
nu_init      = omega_init*t_step;        %[rad] Calc where sc would be

while abs(r_ang - rephase_diff) > tol
    t_init     = t_init + t_step;        %[s]   Begin at first time
step
    nu         = nu + nu_init;           %[rad] Update position in
initial orbit
    mass       = m_init - m_dot*t_step;  %[kg]  Calculate update mass
after thrusting for a time step
    delta_v    = Isp*g0*log(m_init/mass); %[m/s] Calc delta v required
to thrust 1 time step
    m_init     = mass;                   %[kg]  Resets initial mass;
    %Checks if orbit is increasing or decreasing and if it needs to
return
    %to original orbit.
    if check == 1 && t_init < t_max/2
        if rephase_diff > (r_ang)/2
            Vel_new = Vel_new - delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        else
            Vel_new = Vel_new + delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        end
    elseif check == 1 && t_init > t_max/2
        if rephase_diff > (r_ang )/2
            Vel_new = Vel_new - delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        else
            Vel_new = Vel_new - delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        end
    elseif check == 2 && t_init < t_max/2
        if rephase_diff > (r_ang )/2
            Vel_new = Vel_new + delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        else
            Vel_new = Vel_new - delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        end
    elseif check == 2 && t_init > t_max/2
        if rephase_diff > (r_ang)/2
            Vel_new = Vel_new + delta_v*m2km;  %[km/s] Calc new
velocity after delta v
```

```
        else
            Vel_new = Vel_new + delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        end
    end

    r_new     = mu/Vel_new^2;                %[km]   Calc new orbital
radius (assume circular) from new velocity
    if r_new < r_min
        Vel_new = Vel_new - 2*delta_v*m2km;  %[km/s] Calc new velocity
after delta v
        r_new   = mu/Vel_new^2;              %[km]   Calc new orbital
radius (assume circular) from new velocity
    end

    P_new        = 2*pi*sqrt(r_new^3/mu);     %[s]      Period of new
orbit
    omega_new    = 2*pi/P_new;                %[rad/s]  Orbital rate of
new orbit
    nu_new       = nu_old + omega_new*t_step; %[rad]    True anomaly of
the new orbit. This will differ because it will be in a slightly
different orbit than the initial orbit.
    nu_old       = nu_new;
    rephase_diff = abs(nu_new - nu);       %[rad]  Calc difference in
phase angle

    r_new_x(jj)     = r_new*cos(nu_new); %[km]   Calc x-position in
transfer orbit
    r_new_y(jj)     = r_new*sin(nu_new); %[km]   Calc y-position in
transfer orbit
    rephase_tot(jj) = rephase_diff;

    if t_init == t_step
        delta_v_tot(jj) = delta_v_tot + delta_v;       %[km/s] Keeps
track of the total delta v
    else
        delta_v_tot(jj) = delta_v_tot(jj -1) + delta_v; %[km/s] Keeps
track of the total delta v
    end
    %Save variables
    r_new_tr(jj,:)= r_new;                     %[km]
    dv_tot         = delta_v_tot(jj);          %[km/s]
    pos            = [r_new_x; r_new_y];       %[km]
    vel(jj)        = Vel_new;                  %[km/s]
    time(jj)       = t_init;                   %[sec] Keeps track of
time
    jj             = jj+1;                     %[] Count
    prop_mass(jj) = prop_mass(jj -1) + m_dot*t_step; %[kg] Keeps track
of total propellant
    mp_ep          = prop_mass(jj);            %[kg]
    mf             = mass;                     %[kg]
    %Ensure Constraints are met
    if abs(r_new - r_init) < 0.1  &&  t_init >= 50*t_step
        break
    end
```

```
    if t_init >= t_max
        break
    end
end
Th_ep = m_init*dv_tot/time(end);
```

---

chem._rephase_ts.m file

```
%Chemical Rephase Sweep, this program runs chem_rephase for a variety
of time constraints.

%Written by Tiffany Rexius 2010 - 2011

%*********************Constants****************************
mu      = 3.986e5;  %[km^3/s^2]
g0      = 9.81;     %[m/s^2]
r_earth = 6378;     %[km]
hr2sec  = 60*60;    %[sec/hr]
sec2hr  = 1/hr2sec; %[hr/sec]
deg2rad = pi/180;   %[rad/deg]
rad2deg = 180/pi;   %[deg/rad]
km2m    = 1000;     %[m/km]
m2km    = 1/1000;   %[km/m]
%*********************************************************


m_init    = 180;
r_init    = 500 + r_earth;
R_init    = r_init;
Isp       = 235;
Isp_ep    = 600;
Thrust_ep = 0.11;
r_ang     = 180*deg2rad;
nu_init   = 0;
imp       = 20*60;
count     = 1;

min_time = 4*hr2sec;
max_time = 65*hr2sec;
step_size = 1*hr2sec;

for t_tot = min_time:step_size:max_time

    [pos1,pos2,vel,time,mp,dv,nu_diff,Th_ch] =
chem_rephase(m_init,r_init,Isp,r_ang,t_tot,nu_init,imp);

    time_left(count)       = t_tot - time(end);
    trans_trackx(count)    = pos1(1,end);
    trans_tracky(count)    = pos1(2,end);
    init_sat_trackx(count) = pos2(1,end);
    init_sat_trackx(count) = pos2(2,end);
    time_tot(count)        = time(end);
    m_prop(count)          = mp(end);
    dv_tot(count)          = dv(end);
```

```
    rephaseang(count)      = nu_diff(end);
    ave_th(count)          = Th_ch;
    t_track(count)         = t_tot;
    count                  = count + 1;
end

%This section runs the same sweep as was done previously, but has the
%electric system thrust for the "time left" duration, such that the
time
%constraint is very nearly met.

time_dec = 40;
kk       = 1;
jj       = 1;

for t_tot = min_time:step_size:max_time
    t_max        = time_left(kk) - 60;
    rephase_diff = 0;   %Initial rephase amount
    check        = 2;   %1 = decrease alt, 2 = increase alt
    [rephase_tot,nu,pos,vel,t_ep,mf,r_new,dv_tot,Th_ep] =
ep_rephase_time(r_init,m_init,t_max,r_ang,rephase_diff,Isp_ep,Thrust_ep
,check);
    nu_init      = nu(end);
    mp_ep        = m_init - mf(end);
    r_ang2       = r_ang - rephase_tot(end);
    t_chem       = t_tot - t_ep(end);
    t_track(kk)  = t_tot;

    [pos1,pos2,vel,t_ch,mp,dv,nu_diff,Th_ch] =
chem_rephase(mf(end),r_new,Isp,r_ang2,t_chem,nu_init,imp);

    time_left2(kk)       = t_tot - t_ep(end) - t_ch(end);
    t_trans_track(kk)    = t_tot - time_left2(kk);
    trans_trackx2(kk)    = pos1(1,end);
    trans_tracky2(kk)    = pos1(2,end);
    init_sat_trackx2(kk) = pos2(1,end);
    init_sat_trackx2(kk) = pos2(2,end);
    time_tot2(kk)        = t_tot - time_left2(end);
    m_prop2(kk)          = mp(end) + mp_ep;
    dv_tot2(kk)          = dv(end)*km2m + dv_tot(end);
    ave_th2(kk)          = Th_ch;
    rephaseang2(kk)      = nu_diff(end);
    ave_th2              = Th_ch;

    while t_tot(end) > min_time && m_prop2(jj) > m_prop(jj)
        if time_left(kk) < 3*60
            t_chem               = t_tot(end);
            [pos1,pos2,vel,t_ch,mp,dv,nu_diff,Th_ch] =
chem_rephase(m_init, r_init,Isp,r_ang,t_chem,nu_init,imp);
            time_left2(kk)       = t_tot - t_ep(end) - t_ch(end);
            trans_trackx2(kk)    = pos1(1,end);
            trans_tracky2(kk)    = pos1(2,end);
            init_sat_trackx2(kk) = pos2(1,end);
            init_sat_trackx2(kk) = pos2(2,end);
            time_tot2(kk)        = t_tot - time_left2(end);
```

```
            m_prop2(kk)          = mp(end);
            dv_tot2(kk)          = dv(end);
            ave_th2(kk)          = Th_ch;
            rephaseang2(kk)      = nu_diff(end);
            ave_th2              = Th_ch;
        else
            t_max        = time_left(kk) - time_dec;
            if t_max < 0
                t_max = 0;
            end
            r_init       = R_init;
            time_dec     = time_dec + 60;
            rephase_diff = 0;   %Initial rephase amount
            check        = 2;   %1 = decrease alt, 2 = increase alt
            [rephase_tot, nu, pos, vel, t_ep, mf, r_new, dv_tot,Th_ep]
= ep_rephase_time(r_init, m_init, t_max,r_ang, rephase_diff, Isp_ep,
Thrust_ep,check);
            mp_ep        = m_init - mf(end);
            dv_ep        = dv_tot(end);
            m_init       = mf(end);
            nu_init      = nu(end);
            r_ang2       = r_ang - rephase_tot(end);
            r_init2      = sqrt(pos(1,end)^2 + pos(2,end)^2);
            if t_max == 0
                t_chem = t_tot;
            else
            t_chem  = t_tot - t_ep(end);
            end
            [pos1,pos2,vel,t_ch,mp,dv,nu_diff,Th_ch] =
chem_rephase(m_init, r_init2,Isp,r_ang,t_chem,nu_init,imp);
            time_left2(jj)       = t_tot - t_ep(end) - t_ch(end);
            t_trans_track(jj)    = t_tot - time_left2(jj);
            time_taken(jj)       = t_ch + t_ep(end);
            trans_trackx2(jj)    = pos1(1,end);
            trans_tracky2(jj)    = pos1(2,end);
            init_sat_trackx2(jj) = pos2(1,end);
            init_sat_trackx2(jj) = pos2(2,end);
            time_tot2(jj)        = t_tot - time_left2(end);
            m_prop2(jj)          = mp(end) + mp_ep;
            dv_tot2(jj)          = dv(end)*km2m + dv_tot(end);
            ave_th2(jj)          = Th_ch;
            rephaseang2(jj)      = nu_diff(end);
            if m_prop2(jj) < m_prop(jj)
                time_dec = 60;
            end
        end
    end
    if time_left2(kk) > 10*60 && t_tot > max_time/2
        t_max        = t_tot - t_ch - 6*60;
        rephase_diff = 0;   %Initial rephase amount
        check        = 2;   %1 = decrease alt, 2 = increase alt
        [rephase_tot,nu,pos,vel,t_ep,mf,r_new,dv_tot,Th_ep] =
ep_rephase_time(r_init,m_init,t_max,r_ang,rephase_diff,Isp_ep,Thrust_ep
,check);
        nu_init      = nu(end);
        mp_ep        = m_init - mf(end);
```

```
        r_ang2       = r_ang - rephase_tot(end);
        t_chem       = t_tot - t_ep(end);
        t_track(kk)  = t_tot;

        [pos1,pos2,vel,t_ch,mp,dv,nu_diff,Th_ch] =
chem_rephase(mf(end),r_new,Isp,r_ang2,t_chem,nu_init,imp);

        time_left2(kk)       = t_tot - t_ep(end) - t_ch(end);
        t_trans_track(kk)    = t_tot - time_left2(kk);
        trans_trackx2(kk)    = pos1(1,end);
        trans_tracky2(kk)    = pos1(2,end);
        init_sat_trackx2(kk) = pos2(1,end);
        init_sat_trackx2(kk) = pos2(2,end);
        time_tot2(kk)        = t_tot - time_left2(end);
        m_prop2(kk)          = mp(end) + mp_ep;
        dv_tot2(kk)          = dv(end)*km2m + dv_tot(end);
        ave_th2(kk)          = Th_ch;
        rephaseang2(kk)      = nu_diff(end);
        ave_th2              = Th_ch;
    end
    kk = kk + 1;
    jj = jj + 1;
end

figure(7)
plot(time_left2/3600,t_track/3600,time_left/3600,t_track/3600)
title('Time Remaining from Time Constraint for Combined and Chemical
Only')
xlabel('Time Left (hr)')
ylabel('Total Time Constraint (hr)')
legend('Electric - Chemical','Chemical')

figure(8)
title('Combined System Savings')
subplot(3,1,1)
plot(t_track.*sec2hr,m_prop2,'b',t_track.*sec2hr,m_prop,'r')
axis([4 50 0 80])
legend('Electric - Chemical','Chemical')
xlabel('Time (hr)')
ylabel('Propellant Mass (kg)')
subplot(3,1,2)
plot(t_track.*sec2hr,(m_prop - m_prop2))
axis([4 50 0 0.3])
xlabel('Time (hr)')
ylabel('Propellant Mass Savings (kg)')
subplot(3,1,3)
plot(t_track.*sec2hr,dv_tot2)
axis([4 50 0 1500])
xlabel('Time (hr)')
ylabel('Total Delta V (m/s)')

fprintf('Reference Mission 1, 180 deg rephase in 12 hr, return in 12
hr\n')
t_12hr = find(t_track == 12*3600);
```

```
fprintf('Propellant Mass for Combined system =
%6.2f\n',m_prop2(t_12hr)*2)
fprintf('Delta V for Combined system          =
%6.2f\n',dv_tot2(t_12hr)*2)
fprintf('Transfer Time for Combined system    =
%6.2f\n',t_trans_track(t_12hr)*sec2hr)
```

---

ep_rephase_time.m file

```
function [rephase_tot, nu, pos, vel, time, mf, r_new, dv_tot, Th_ep] =
ep_rephase_time(r_init, m_init, t_max,r_ang, rephase_diff, Isp_ep,
Thrust_ep,check)

%EP will thrust up until time constraint (t_max) is reached.
%No figures from this file.

%Written by Tiffany Rexius 2010-2011
%*********************Constants*****************************
mu      = 3.986e5;  %[km^3/s^2]
g0      = 9.81;     %[m/s^2]
r_earth = 6378;     %[km]
hr2sec  = 60*60;    %[sec/hr]
sec2hr  = 1/hr2sec; %[hr/sec]
deg2rad = pi/180;   %[rad/deg]
rad2deg = 180/pi;   %[deg/rad]
km2m    = 1000;     %[m/km]
m2km    = 1/1000;   %[km/m]
%***********************************************************
Isp         = Isp_ep;
m_dot       = Thrust_ep/(Isp*g0);           %[kg/s]
a_init      = r_init;                       %[km] if circular orbit
P_init      = 2*pi*sqrt(a_init^3/mu);       %[s]
omega_init = 2*pi/P_init;                   %[rad/s]
Vel_init    = sqrt(mu/r_init);              %[km/s]

r_min         = 290 + r_earth;
t_step        = 60;
t_init        = 0;
delta_v_tot   = 0;
prop_mass     = 0;
nu            = 0;
jj            = 1; %Count
Vel_new       = Vel_init;
r_new         = r_init;
nu_old        = 0;
nu_init       = omega_init*t_step; %[rad] Calc. true anomaly where
satellite would be in initial orbit

while t_init < t_max
    t_init      = t_init + t_step;          %[s]   Begin at first time
step
```

```
    nu         = nu + nu_init;            %[rad] Update position in
initial orbit
    mass       = m_init - m_dot*t_step;   %[kg]  Calculate update mass
after thrusting for a time step
    delta_v    = Isp*g0*log(m_init/mass); %[m/s] Calc delta v required
to thrust 1 time step
    m_init     = mass;                    %[kg]  Resets initial mass;

    %If over half of the rephase is completed and the time constraint
is not half over, stay in orbit and stop thrusting.

    if rephase_diff > (r_ang)/2 && t_init < t_max/2
        delta_v = 0;
        m_dot   = 0;
    else
        m_dot   = Thrust_ep/(Isp*g0);
    end

    %Checks if orbit is increasing or decreasing and if it needs to
return to original orbit.
    if check == 1 && t_init < t_max/2
        if rephase_diff > (r_ang)/2
            Vel_new = Vel_new - delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        else
            Vel_new = Vel_new + delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        end
    elseif check == 1 && t_init > t_max/2
        if rephase_diff > (r_ang)/2
            Vel_new = Vel_new - delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        else
            Vel_new = Vel_new + delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        end
    elseif check == 2 && t_init < t_max/2
        if rephase_diff > (r_ang)/2
            Vel_new = Vel_new + delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        else
            Vel_new = Vel_new - delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        end
    elseif check == 2 && t_init > t_max/2
        if rephase_diff > (r_ang)/2
            Vel_new = Vel_new + delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        else
            Vel_new = Vel_new - delta_v*m2km;  %[km/s] Calc new
velocity after delta v
        end
    end
```

```
    r_new     = mu/Vel_new^2;                %[km]   Calc new orbital
radius (assume circular) from new velocity
    if r_new < r_min
        Vel_new = Vel_new - 2*delta_v*m2km;  %[km/s] Calc new velocity
after delta v
        r_new     = mu/Vel_new^2;            %[km]   Calc new orbital
radius (assume circular) from new velocity
    end

    P_new        = 2*pi*sqrt(r_new^3/mu);     %[s]      Period of new
orbit
    omega_new    = 2*pi/P_new;                %[rad/s]  Orbital rate of
new orbit
    nu_new       = nu_old + omega_new*t_step; %[rad]    True anomaly of
the new orbit. This will differ because it will be in a slightly
different orbit than the initial orbit.
    nu_old       = nu_new;
    rephase_diff = abs(nu_new - nu);       %[rad]  Calc difference in
phase angle

    r_new_x(jj)     = r_new*cos(nu_new); %[km]   Calc x-position in
transfer orbit
    r_new_y(jj)     = r_new*sin(nu_new); %[km]   Calc y-position in
transfer orbit
    rephase_tot(jj) = rephase_diff;

    if t_init == t_step
        delta_v_tot(jj) = delta_v_tot + delta_v;   %[km/s] Keeps track
of the total delta v
    else
        delta_v_tot(jj) = delta_v_tot(jj -1) + delta_v;    %[km/s]
Keeps track of the total delta v
    end
    %Save variables
    r_new_tr(jj,:)  = r_new;
    dv_tot          = delta_v_tot(jj);
    pos             = [r_new_x; r_new_y];
    vel(jj)         = Vel_new;
    time(jj)        = t_init;   %[s]    Keeps track of time
    jj              = jj+1;
    prop_mass(jj) = prop_mass(jj -1) + m_dot*t_step; %[kg]   Keeps
track of total propellant
    mp_ep           = prop_mass(jj);
    mf              = mass;
    %Ensure Constraints are met
    if abs(r_new - r_init) < 0.1  &&  t_init >= 50*t_step
        break
    end
end
Th_ep = m_init*dv_tot/time(end);
```

Appendix B: Mission 2 MATLAB Code

Four MATLAB files were written to model the altitude raise and lower mission. The primary file is Run_M2.m. This file is where all the initial mission parameters are entered. A change to this file is reflected throughout all the calculations. The files that specifically calculate the altitude change are chem_maneuver.m and ep_maneuver.m. These files are accessed in orb_raise_ts.m which runs numerous thrust profiles. The order the scripts are listed in this file are:

1) Run_M2.m

2) orb_raise_ts.m

3) chem_maneuver.m

4) ep_maneuver.m

Run_M2.m file

```
%Mission 2: This program raises the initial orbit of 500km to 1500km
%if the ep system takes too long (> 48hr), it uses ep for the majority
of the way, then switches to chemical to make the time constraint.
%The program then computes the time it would take to return to the
%original orbit with EP.

%This program runs - orb_raise_ts.m, ep_manuever.m, chem_manuever.m

%Written by Tiffany Rexius 2010 - 2011
clear;
clc;
%****************Vehicle Information*************************
m_init      = 180;  %[kg]
P_avail     = 1000; %[W]
Isp_ep      = 600;  %[s]
Thrust_ep   = 0.11; %[N]
Isp_chem    = 235;  %[s]
veh_prop = [m_init, Isp_ep, Thrust_ep, Isp_chem, P_avail];
%************************************************************
%*********************Constants*****************************
mu      = 3.986e5;  %[km^3/s^2]
g0      = 9.81;     %[m/s^2]
r_earth = 6378;     %[km]
hr2sec  = 60*60;    %[sec/hr]
```

```
sec2hr  = 1/hr2sec; %[hr/sec]
deg2rad = pi/180;   %[rad/deg]
rad2deg = 180/pi;   %[deg/rad]
km2m    = 1000;     %[m/km]
m2km    = 1/1000;   %[km/m]
%*************************************************************
%***********************Mission 2**************************
t_const     = 48*hr2sec;           %[sec]
rad_init    = 500;                 %[km]
rad_final   = 1500;                %[km]
imp         = 20*60;               %[sec] Impulsive burn for 20 minutes
r_init      = rad_init + r_earth;  %[km]
r_final     = rad_final + r_earth; %[km]
rephase_ang = 0;                   %[rad]
t_rtn       = 30*24*hr2sec;        %[sec] Return trip time
%*************************************************************
%Tradespace Generator for EP 0% of time to 98% of time*******
[t_ep,t_ch,Th_ch,Th_ep,ec_var,cec_var,ece_var,ece2_var,ce_var] =
orb_raise_ts(r_init,r_final,t_const,veh_prop,imp);

mp_ec     = ec_var(1,:);
dv_ec     = ec_var(2,:);
t_ec      = ec_var(3,:);
Ave_th_ec = ec_var(4,:);
r_ec      = ec_var(5,:);

mp_ce     = ce_var(1,:);
dv_ce     = ce_var(2,:);
t_ce      = ce_var(3,:);
Ave_th_ce = ce_var(4,:);
r_ce      = ce_var(5,:);

mp_cec     = cec_var(1,:);
dv_cec     = cec_var(2,:);
t_cec      = cec_var(3,:);
Ave_th_cec = cec_var(4,:);
r_cec      = cec_var(5,:);

mp_ece     = ece_var(1,:);
dv_ece     = ece_var(2,:);
t_ece      = ece_var(3,:);
Ave_th_ece = ece_var(4,:);
r_ece      = ece_var(5,:);

mp_ece2     = ece2_var(1,:);
dv_ece2     = ece2_var(2,:);
t_ece2      = ece2_var(3,:);
Ave_th_ece2 = ece2_var(4,:);
r_ece2      = ece2_var(5,:);

%**************Electric Propulsion System*******************
%For return trip time constraint is 30 days
[pos_ep,vel_ep,time_ep,mf,mf_ep,nu,dv_ep,Th_ep]=
ep_maneuver(r_init,r_final,t_rtn,Isp_ep,Thrust_ep,m_init,P_avail);
```

```
dv_ep_only  = dv_ep;      %[km/s]
mp_ep_only  = mf_ep;      %[kg]
t_ep_only   = time_ep;    %[sec]
Th_ep_eonly = Th_ep;      %[N]
%*************************************************************
%**************Chemical Propulsion System******************
nu = 0;
[pos_chem,vel_chem,time_chem,mf_chem,dv_chem,Th_ch] =
chem_maneuver(r_init,r_final,Isp_chem,m_init,nu,t_const,imp);

dv_ch_only   = dv_chem*km2m; %[km/s]
mp_ch_only   = mf_chem;      %[kg]
t_ch_only    = time_chem;    %[sec]
ave_th_conly = Th_ch;        %[N] Average Chemical Thrust for combined
%*************************************************************
%**********Electric/Chemical Propulsion System**************
r_init2 = r_init;
if time_ep(end) > t_const
    t_max_ep = t_const - 0.25*hr2sec;
    t_pos    = find(time_ep == t_max_ep);
    r_x      = pos_ep(1,t_pos);
    r_y      = pos_ep(2,t_pos);
    r_final2 = sqrt(r_x(1)^2+r_y(1)^2);

    [pos_ep,vel_ep,time_ep,mf,mf_ep,nu,dv_ep,Th_ep] =
ep_maneuver(r_init2,r_final2,t_const,Isp_ep,Thrust_ep,m_init,P_avail);
    r_final_ep   = sqrt(pos_ep(1,end)^2+pos_ep(2,end)^2);
    ave_th_ecomb = Th_ep;
    r_init2      = r_final2;

    [pos_chem,vel_chem,time_chem,mf_chem,dv_chem,Th_ch] =
chem_maneuver(r_final_ep,r_final,Isp_chem,mf,nu,t_const,imp);
    time_tot    = time_ep(end)*sec2hr + time_chem(end);
    ave_th_comb = Th_ch;   %(N) Average Chemical Thrust for combined
end
r_ep_check   = sqrt(pos_ep(1,:).^2 + pos_ep(2,:).^2);
ave_th_tot = (ave_th_ecomb*time_ep(end) +
ave_th_comb*imp*2)/(time_ep(end) + imp*2);
%*************************************************************
%*Plot outline of earth on plot and outline of initial orbit*
nu_earth = 0;
r_earth  = 6378;    %km
nu_step  = 1*pi/180;
kk = 1;
while nu_earth < 2*pi
    r_earth_x(kk) = r_earth*cos(nu_earth);
    r_earth_y(kk) = r_earth*sin(nu_earth);
    r_init_x(kk)  = (r_init)*cos(nu_earth);
    r_init_y(kk)  = (r_init)*sin(nu_earth);
    r_final_x(kk) = (r_final)*cos(nu_earth);
    r_final_y(kk) = (r_final)*sin(nu_earth);
    nu_earth      = nu_earth + nu_step;
    kk            = kk+1;
end
```

```
figure(1)
plot(mp_ec,t_ec,'b',mp_cec,t_cec,'r',mp_ece,t_ece,'g',mp_ece2,t_ece2,'-
^r',mp_ce,t_ce,'-xb')
title('Propellant Mass vs. Trip Time for 100% Chemical to 2% Chemical')
legend('Electric - Chemical','Chemical - Electric - Chemical','Electric
- Chemical - Electric','Electric - Chemical - Electric remaining
time','Chemical - Electric')
xlabel('Propellant Mass')
ylabel('Total Trip time (hr)')
ts = linspace(5,90,length(t_ep));

figure(2)
plot(pos_ep(1,:),pos_ep(2,:),r_init_x,r_init_y,r_earth_x,r_earth_y,pos_
chem(1,:),pos_chem(2,:),r_final_x, r_final_y)
legend('Transfer Orbit EP [km]','Initial Orbit [km]','Earth
[km]','Transfer Orbit Chemical [km]','Final Orbit [km]')

figure(3)
plot(mp_ep_only,t_ep_only(end)*sec2hr,'o',...
    mp_ch_only,t_ch_only(end),'^',...
    mp_ec(end),t_ec(end),'x',...
    mp_ece2(end),t_ece2(end),'d',...
    mp_ece(end),t_ece(end),'+',...
    mp_ce(end),t_ce(end),'s',...
    mp_cec(end),t_cec(end),'<','MarkerSize',10)
grid on
ylabel('Trip time (hr)')
xlabel('Propellant Mass (kg)')
legend('EP Only System',...
    'Chemical Only System',...
    'Electric - Chemical',...
    '98% of Time Electric - Chemical - Electric',...
    '49% of Time Electric, Chemical, 49% of time Electric',...
    'Chemical - 98% of Time Electric',...
    'Chemical - 98% of Time Electric - Chemical')

figure(4)
plot(t_ec,Ave_th_ec,'b',t_cec,Ave_th_cec,'r',t_ece,Ave_th_ece,'g',t_ece
2,Ave_th_ece2,'-^r',t_ce,Ave_th_ce,'-xb')
title('Mission 2: 1000 km orbit raise')
legend('Electric - Chemical','Chemical - Electric - Chemical','Electric
- Chemical - Electric','Electric - Chemical - Electric remaining
time','Chemical - Electric')
ylabel('Average Thrust (N)')
xlabel('% Electric Propulsion')

figure(5)
plot(mp_cec,t_cec,'x')
title('Chemical/Electric/Chemical Propulsion from 100% Chemical to 5%
Chemical')
ylabel('Time to Complete Mission (hr)')
xlabel('Propellant Mass to Complete Mission (kg)')

figure(6)
subplot(2,1,1)
```

```
plot(Ave_th_ec,mp_ec,'b',Ave_th_cec,mp_cec,'r',Ave_th_ece,mp_ece,'g',Av
e_th_ece2,mp_ece2,'-^r',Ave_th_ce,mp_ce,'-xb')
title('Mission 2: 1000 km orbit raise for 100% Chemical to 2%
Chemical')
legend('Electric - Chemical','Chemical - Electric - Chemical','Electric
- Chemical - Electric','Electric - Chemical - Electric remaining
time','Chemical - Electric')
xlabel('Average Thrust (N)')
ylabel('Propellant Mass (kg)')
subplot(2,1,2)
plot(Ave_th_ec,t_ec,'b',Ave_th_cec,t_cec,'r',Ave_th_ece,t_ece,'g',Ave_t
h_ece2,t_ece2,'-^r',Ave_th_ce,t_ce,'-xb')
title('Mission 2: 1000 km orbit raise for 100% Chemical to 2%
Chemical')
legend('Electric - Chemical','Chemical - Electric - Chemical','Electric
- Chemical - Electric','Electric - Chemical - Electric remaining
time','Chemical - Electric')
xlabel('Average Thrust (N)')
ylabel('Trip Time (hr)')

figure(7)
plot(linspace(0,98,21),mp_ece,'-x',linspace(0,98,21),mp_ece2,'-
^',linspace(0,98,21),mp_ec,'-<',linspace(0,98,21),mp_cec,'-
o',linspace(0,98,21),mp_ce,'-s')
legend('Electric - Chemical - Electric','Electric - Chemical - Electric
remaining time','Electric - Chemical','Chemical - Electric -
Chemical','Chemical - Electric')
xlabel('% Electric Propulsion')
ylabel('Propellant Mass (kg)')

fprintf('\n\n*******Results for Mission 2: Orbit Raise*********/n')
fprintf('   Initial orbit (km) = %6.2f\n',rad_init)
fprintf('   Final orbit (km)   = %6.2f\n\n',rad_final)
fprintf('******Chemical System Only*******      ******Electric System
Only******\n')
fprintf('Transfer time (hr)   = %6.2f          Transfer time (hr)   =
%6.2f\n',t_ch_only,t_ep_only(end)*sec2hr)
fprintf('Delta V (m/s)        = %6.2f          Delta V (m/s)        =
%6.2f\n',dv_ch_only,dv_ep_only)
fprintf('Propellant Mass (kg) = %6.2f          Propellant Mass (kg) =
%6.2f\n',mp_ch_only,mp_ep_only)
fprintf('Average Thrust(N)    = %6.2f          Average Thrust (N)   =
%6.2f\n\n',ave_th_conly,Thrust_ep)
fprintf('*Electric/Chemical System Values*      *Chemical/Electric
System Values*\n')
fprintf('Trip Time (hr)       = %6.2f          Trip Time (hr)       =
%6.2f\n',t_ec(end),t_ce(end))
fprintf('Delta V (m/s)        = %6.2f          Delta V (m/s)        =
%6.2f\n',dv_ec(end),dv_ce(end))
fprintf('Propellant Mass (kg) = %6.2f          Propellant Mass (kg) =
%6.2f\n',mp_ec(end),mp_ce(end))
fprintf('Average Thrust(N)    = %6.2f          Average Thrust(N)    =
%6.2f\n\n',Ave_th_ec(end),Ave_th_ce(end))
fprintf('*Elec/Chemical/Elec System Values*     *Elec/Chemical/Elec2
System Values*\n')
```

```
fprintf('Trip Time (hr)       = %6.2f          Trip Time (hr)       =
%6.2f\n',t_ece(end),t_ece2(1))
fprintf('Delta V (m/s)        = %6.2f          Delta V (m/s)        =
%6.2f\n',dv_ece(end),dv_ece2(1))
fprintf('Propellant Mass (kg) = %6.2f          Propellant Mass (kg) =
%6.2f\n',mp_ece(end),mp_ece2(1))
fprintf('Average Thrust(N)    = %6.2f          Average Thrust(N)    =
%6.2f\n\n',Ave_th_ece(end),Ave_th_ece2(1))
fprintf('*Chemical/Electric/Chemical System Values*\n')
fprintf('Trip Time (hr)       = %6.2f\n',t_cec(end))
fprintf('Delta V (m/s)        = %6.2f\n',dv_cec(end))
fprintf('Propellant Mass (kg) = %6.2f\n',mp_cec(end))
fprintf('Average Thrust(N)    = %6.2f\n\n',Ave_th_cec(end))
fprintf('*********Orbit Raise & Return*************\n')
fprintf('***EP thrust initially, then Chem*********\n')
fprintf('Trip Time(hr)        = %6.2f\n',t_ece(end))
fprintf('Delta V (m/s)        = %6.2f\n',dv_ece(end) + dv_ep_only)
fprintf('Propellant Mass (kg) = %6.2f\n',mp_ece(end)+ mp_ep_only)
fprintf('Average Thrust(N)    = %6.2f\n\n',(Ave_th_ece(end) +
Th_ep_eonly)/2)
```

---

ep_maneuver.m file

```
function [pos,vel,time,mf,mp_ep,nu,dv_tot,Th_ep] =
ep_maneuver(r_init,r_final,t_max,Isp,Thrust,m_init,P_avail)
format long
%***********Constants*******************************************
mu      = 3.986e5; %[km^3/s^2]
g0      = 9.81;    %[m/s^2]
r_earth = 6378;    %[km]
hr2sec  = 60*60;   %[sec]
m2km    = 1/1000;  %[km]
%**************************************************************

m_dot      = Thrust/(Isp*g0);          %[kg/s]
a_init     = r_init;                   %[km] if circular orbit
P_init     = 2*pi*sqrt(a_init^3/mu);   %[s]
omega_init = 2*pi/P_init;              %[rad/s]
Vel_init   = sqrt(mu/r_init);          %[km/s]

t_step       = 90;       %[s]
t_init       = 0;        %[s]
delta_v_tot  = 0;        %[km/s]
prop_mass    = 0;        %[kg]
nu           = 0;        %[rad]
jj           = 1;        %[] Count
Vel_new      = Vel_init; %[km/s]
nu_old       = 0;        %[rad]
tol          = 0.2;      %[km]
mass         = m_init;   %[kg]
```

```
while abs(r_final - r_init) > tol
    t_init     = t_init + t_step;       %[s]   Begin at first time step
    nu_init    = omega_init*t_step;     %[rad] Calc. true anomaly where
satellite would be in initial orbit
    nu         = nu + nu_init;          %[rad] Update position in
initial orbit
    mass       = mass - m_dot*t_step;   %[kg]  Calculate update mass
after thrusting for a time step

    delta_v    = Isp*g0*log(m_init/mass); %[m/s] Calc delta v required
to thrust 1 time step

    m_init     = mass;                    %[kg]  Resets initial mass;

    if r_final < r_init
        Vel_new = Vel_new + delta_v*m2km;    %[km/s] Calc new velocity
after delta v
    else
        Vel_new = Vel_new - delta_v*m2km;    %[km/s] Calc new velocity
after delta v
    end

    r_new      = mu/Vel_new^2;              %[km]   Calc new orbital
radius (assume circular) from new velocity
    P_new      = 2*pi*sqrt(r_new^3/mu);     %[s]    Period of new orbit
    omega_new  = 2*pi/P_new;                %[rad/s] Orbital rate of
new orbit
    nu_new     = nu_old + omega_new*t_step; %[rad] True anomaly of the
new orbit. This will differ because it will be in a slightly different
orbit than the initial orbit.
    nu_old     = nu_new;

    r_new_x(jj)  = r_new*cos(nu_new);      %[km]   Calc x-position in
transfer orbit
    r_new_y(jj)  = r_new*sin(nu_new);      %[km]   Calc y-position in
transfer orbit

    r_init       = r_new;

    time_total(jj) = t_init;               %[s]    Keeps track of time

    if t_init == t_step
        delta_v_tot(jj) = delta_v;
    else
        delta_v_tot(jj) = delta_v_tot(jj-1) + delta_v;
    end
    nu_track(jj) = nu;
    vel(jj)      = Vel_new;
    time(jj)     = t_init;
    dv_tot       = delta_v_tot(jj);
    jj           = jj+1;
    prop_mass(jj)= prop_mass(jj -1) + m_dot*t_step; %[kg] Keeps track
of total propellant
    mp_ep        = prop_mass(jj);
    if t_init > t_max
```

```
        break
    end
end
Th_ep = m_init*dv_tot/(time(end)); %[N]
pos   = [r_new_x; r_new_y];
nu    = nu_old;
mf    = m_init;
```

---

chem_maneuver.m file

```
function [pos,vel,time,m_prop,dv_tot,Th_ch]=
chem_maneuver(r_init,r_final,Isp,m_init,nu,t_tot,imp)
format long

%***********Constants*******************************************
mu      = 3.986e5; %km^3/s^2
g0      = 9.81;    %m/s^2
sec2hr  = 1/3600;
km2m    = 1000;
%*******************************************************************
burn_time    = 2*imp;                        %[sec] for Hohmann 2 burns
a_init       = r_init;                       %[km] if circular orbit
P_init       = 2*pi*sqrt(a_init^3/mu);       %[s]
omega_init   = 2*pi/P_init;                  %[rad/s]
Vel_init     = sqrt(mu/r_init);              %[km/s]

a_trans      = (r_init + r_final)/2;           %[km]
P_trans      = 2*pi*sqrt(a_trans^3/mu);        %[s]
omega_trans  = 2*pi/P_trans;                   %[rad/s]
Energy_tr    = -mu/(2*a_trans);                %[km^2/s^2]
Vel_trans1   = sqrt(2*(mu/r_init+Energy_tr)); %[km/s]

dv1          = abs(Vel_init - Vel_trans1);      %[km/s]
time         = pi*sqrt(a_trans^3/mu)*sec2hr;    %[s]

Vel_final    = sqrt(mu/r_final);                %[km/s]
a_final      = r_final;                         %[km]
P_final      = 2*pi*sqrt(a_final^3/mu);         %[s]
omega_final  = 2*pi/P_final;                    %[rad/s]
Vel_trans2   = sqrt(2*(mu/r_final+Energy_tr)); %[km/s]
dv2          = abs(Vel_final - Vel_trans2);     %[km/s]
dv_tot       = dv2 + dv1;
m_prop       = m_init - m_init*exp(-dv_tot*km2m/Isp/g0);
mf           = m_init - m_prop;
Th_ch        = m_init*dv_tot*km2m/(burn_time); %[N]
a            = (r_init + r_final)/2;
ecc          = 1 - r_init/a;
per          = a*(1 - ecc^2);

nu_init      = nu;
nu_final     = nu + 2*pi/2; %[rad] Hohmann transfer so 180 deg
```

```
kk            = 1;

while nu < nu_final
    r_trans      = per/(1+ecc*cos(nu - nu_init));
    v_trans(kk)  = sqrt(2*(Energy_tr + mu/r_trans));
    r_init_x(kk) = r_trans*cos(nu - nu_init);
    r_init_y(kk) = r_trans*sin(nu - nu_init);
    nu           = nu + 1*pi/180;
    kk           = kk+1;
end
pos = [r_init_x; r_init_y];
vel = v_trans;
```

---

orb_raise_ts file

```
function
[t_ep,t_ch,Th_ch_tr,Th_ep_tr,ec_var,cec_var,ece_var,ece2_var,ce_var] =
orb_raise_ts(r_init,r_final,t_const,veh_prop,imp)

m_init  = veh_prop(1);
Isp_ep  = veh_prop(2);
Thr_ep  = veh_prop(3);
Isp_ch  = veh_prop(4);
P_avail = veh_prop(5);

%***********************Constants*****************************
mu      = 3.986e5;  %[km^3/s^2]
g0      = 9.81;     %[m/s^2]
r_earth = 6378;     %[km]
hr2sec  = 60*60;    %[sec/hr]
sec2hr  = 1/hr2sec; %[hr/sec]
deg2rad = pi/180;   %[rad/deg]
rad2deg = 180/pi;   %[deg/rad]
km2m    = 1000;     %[m/km]
m2km    = 1/1000;   %[km/m]
%************************************************************
min_ep  = 0;
max_ep  = 0.981;
step_ep = max_ep/20;
%*************************************************************
kk = 1;
for nu = 0:0.01:2*pi
    rx_init(kk)  = r_init*cos(nu);
    ry_init(kk)  = r_init*sin(nu);
    rx_final(kk) = r_final*cos(nu);
    ry_final(kk) = r_final*sin(nu);
    kk           = kk + 1;
end

%*********************************************************************
%**********Electric/Chemical Propulsion System**********************
```

```
%**********************************************************************
kk = 1;
for i = min_ep:step_ep:max_ep
    t_max_ep     = t_const*(i);
    [pos_ep,vel_ep,time_ep,mf,mp_ep,nu,dv_ep,Th_ep] =
ep_maneuver(r_init,r_final,t_max_ep,Isp_ep,Thr_ep,m_init,P_avail);
    r_init2      = sqrt(pos_ep(1,end)^2+pos_ep(2,end)^2);
    t_ep(kk)     = time_ep(end);
    Th_ep_tr(kk) = Th_ep;
    t_ch_cons    = t_const - time_ep(end);
    rx_ep1     = pos_ep(1,:);
    ry_ep1     = pos_ep(2,:);

    [pos_ch,vel_ch,time_ch,mf_ch,dv_ch,Th_ch] =
chem_maneuver(r_init2,r_final,Isp_ch,mf,nu,t_ch_cons,imp);
    rx_ch1        = pos_ch(1,:);
    ry_ch1        = pos_ch(2,:);
    r_ec(kk)      = sqrt(pos_ch(1,end)^2 + pos_ch(2,end)^2);
    Th_ch_tr(kk)  = Th_ch;
    t_ch(kk)      = time_ch;
    Ave_th_ec(kk) =
(Th_ch*time_ch*hr2sec+Th_ep*time_ep(end))/(time_ep(end) +
time_ch*hr2sec);
    mp_ec(kk)     = mf_ch + mp_ep;
    t_ec(kk)      = time_ep(end)*sec2hr + time_ch(end);
    dv_ec(kk)     = dv_ch*km2m + dv_ep;
    kk            = kk +1;
end
figure(12)
plot(rx_ch1,ry_ch1,'-x',rx_ep1,ry_ep1,'--
',rx_init,ry_init,rx_final,ry_final)
title('Electric - Chemical Profile')
legend('Electric Segment','ChemicalSegment','Initial Orbit','Final
Orbit')

%*********************************************************
%DEVELOPING ANOTHER THRUST SCHEME, 22 Sep 10
%***********Chemical/Electric/Chemical Propulsion System**************
jj = 1;
max_ep2 = max_ep - step_ep*0.39;
step_ep2 = max_ep2/20;
for i = min_ep:step_ep2:max_ep2
    t_ep1       = i*t_const;
    [pos_ep1,vel_ep1,time_ep1,mf1,mp_ep1,nu1,dv_ep1,Th_ep1] =
ep_maneuver(r_init + 500,r_final,t_ep1,Isp_ep,Thr_ep,m_init,P_avail);
    %Find what altitude increase can be achieved by EP in time period
    %Use Chemical to perform half of the remaining altitude change
    r_ep_final = sqrt(pos_ep1(1,end)^2+pos_ep1(2,end)^2);
    r_ep       = r_ep_final - r_init;
    r_ch_seg   = r_ep/2 + r_init;
    nu         = 0;

    [pos_ch1,vel_ch1,time_ch1,mf_ch1,dv_ch1,Th_ch1] =
chem_maneuver(r_init,r_ch_seg,Isp_ch,m_init,nu,t_const,imp);
    r_init_ep  = sqrt(pos_ch1(1,end)^2+pos_ch1(2,end)^2);
    m_init_ep  = m_init - mf_ch1;
```

```
    rx_ch1     = pos_ch1(1,:);
    ry_ch1     = pos_ch1(2,:);

    [pos_ep2,vel_ep2,time_ep2,mf2,mp_ep2,nu2,dv_ep2,Th_ep2] =
ep_maneuver(r_init_ep,r_final,t_ep1,Isp_ep,Thr_ep,m_init_ep,P_avail);
    rx_ep1    = pos_ep2(1,:);
    ry_ep1    = pos_ep2(2,:);
    r_init_ch = sqrt(pos_ep2(1,end)^2+pos_ep2(2,end)^2);
    t_ch_sec  = t_const - time_ep2 - time_ch1*hr2sec;

    [pos_ch2,vel_ch2,time_ch2,mf_ch2,dv_ch2,Th_ch2] =
chem_maneuver(r_init_ch,r_final,Isp_ch,mf,nu2,t_ch_sec,imp);
    rx_ch2         = pos_ch2(1,:);
    ry_ch2         = pos_ch2(2,:);
    Ave_th_cec(jj) = (Th_ch2*time_ch1*hr2sec +
Th_ch1*time_ch1*hr2sec+Th_ep2*time_ep2(end))/(time_ep2(end) + (time_ch2
+ time_ch1)*hr2sec);
    r_cec(jj)      = sqrt(pos_ch2(1,end)^2+pos_ch2(2,end)^2);
    mp_cec(jj)     = mf_ch1 + mf_ch2 + mp_ep;
    dv_cec(jj)     = dv_ch1*km2m + dv_ep(end) + dv_ch2*km2m;
    t_cec(jj)      = time_ch1 + time_ch2 + time_ep2(end)*sec2hr;
    jj             = jj + 1;
end
% figure(8)
% plot(rx_ch1,ry_ch1,'-x',rx_ch2,ry_ch2,'-x',rx_ep1,ry_ep1,'--
',rx_init,ry_init,rx_final,ry_final)
% title('Chemical - Electric - Chemical Profile')
% legend('1st Chemical Segment','Electric Segment','2nd Chemical
Segment','Initial Orbit','Final Orbit')
%*********************************************************
%DEVELOPING ANOTHER THRUST SCHEME, 28 Sep 10
%**********Electric/Chemical/Electric Propulsion System**************
kk = 1;
for i = min_ep:step_ep:max_ep
    t_ep1       = i*t_const/2;
    [pos_ep1,vel_ep1,time_ep1,mf1,mp_ep1,nu1,dv_ep1,Th_ep1] =
ep_maneuver(r_init,r_final,t_ep1,Isp_ep,Thr_ep,m_init,P_avail);

    %Find what altitude increase can be achieved by EP in time period
    r_ep_final = sqrt(pos_ep1(1,end)^2+pos_ep1(2,end)^2);
    r_ep       = r_ep_final - r_init;
    r_ep_final = r_init + 1.2*r_ep;
    r_ch_seg   = ((r_final - r_init) - (r_ep_final - r_init)) + r_init;

    [pos_ch1,vel_ch1,time_ch1,mf_ch1,dv_ch1,Th_ch1] =
chem_maneuver(r_ep_final,r_ch_seg,Isp_ch,m_init,nu1,t_const,imp);

    r_init_ep  = sqrt(pos_ch1(1,end)^2+pos_ch1(2,end)^2);
    m_init_ep  = m_init - mf_ch1;

    while abs(r_init_ep - r_final) < 0.2
        r_ch_seg   = r_ch_seg - 1;
        [pos_ch1,vel_ch1,time_ch1,mf_ch1,dv_ch1,Th_ch1] =
chem_maneuver(r_ep_final,r_ch_seg,Isp_ch,m_init,nu1,t_const,imp);
        r_init_ep  = sqrt(pos_ch1(1,end)^2+pos_ch1(2,end)^2);
```

```
        m_init_ep  = m_init - mf_ch1;
    end

    [pos_ep2,vel_ep2,time_ep2,mf2,mp_ep2,nu2,dv_ep2,Th_ep2] = 
ep_maneuver(r_init_ep,r_final,t_ep1,Isp_ep,Thr_ep,m_init_ep,P_avail);

    while time_ep2(end) < t_ep1
        r_ch_seg   = r_ch_seg - 1;
        [pos_ch1,vel_ch1,time_ch1,mf_ch1,dv_ch1,Th_ch1] = 
chem_maneuver(r_ep_final,r_ch_seg,Isp_ch,m_init,nu1,t_const,imp);
        r_init_ep  = sqrt(pos_ch1(1,end)^2+pos_ch1(2,end)^2);
        m_init_ep  = m_init - mf_ch1;
        [pos_ep2,vel_ep2,time_ep2,mf2,mp_ep2,nu2,dv_ep2,Th_ep2] = 
ep_maneuver(r_init_ep,r_final,t_ep1,Isp_ep,Thr_ep,m_init_ep,P_avail);
    end
    rx_ep1          = pos_ep1(1,:);
    ry_ep1          = pos_ep1(2,:);
    rx_ep2          = pos_ep2(1,:);
    ry_ep2          = pos_ep2(2,:);
    rx_ch1          = pos_ch1(1,:);
    ry_ch1          = pos_ch1(2,:);
    Ave_th_ece(kk) = (Th_ep2*time_ep2(end) + 
Th_ch1*time_ch1*hr2sec+Th_ep1*time_ep1(end))/(time_ep2(end) + 
time_ep1(end) + time_ch1*hr2sec);
    r_ece(kk)       = sqrt(pos_ep2(1,end)^2+pos_ep2(2,end)^2);
    mp_ece(kk)      = mf_ch1 + mp_ep1 + mp_ep2;
    dv_ece(kk)      = dv_ch1*km2m + dv_ep1(end) + dv_ep2(end);
    t_ece(kk)       = time_ch1 + time_ep1(end)*sec2hr + 
time_ep2(end)*sec2hr;
    kk               = kk + 1;

end

% figure(9)
% plot(rx_ep1,ry_ep1,'--',rx_ch1,ry_ch1,'-x',rx_ep2,ry_ep2,'--
',rx_init,ry_init,rx_final,ry_final)
% title('Electric - Chemical - Elecric Profile')
% legend('1st Electric Segment','Chemical Segment','2nd Electric 
Segment','Initial Orbit','Final Orbit')
%**********************************************************
%DEVELOPING ANOTHER THRUST SCHEME, 28 Sep 10
%******Electric/Chemical/Electric2 Propulsion System************
%Initial Electric for x% of total time, chemical, electric for 
remainder of time constraint.
kk = 1;
max_ep2 = max_ep - 2*step_ep;
step_ep2 = max_ep2/20;
for i = min_ep:step_ep2:max_ep2
    %Calculate amount of time EP can thrust for, from 0 - 98% of time
    if i == min_ep
        t_ep1    = 0;
        time_ep1 = 0;
        pos_ep1  = [r_init; 0];
        mf1      = m_init;
        mp_ep1   = 0;
```

```
        nu1      = 0;
        dv_ep1   = 0;
        Th_ep1   = 0;
    else
        t_ep1       = i*t_const;
        [pos_ep1,vel_ep1,time_ep1,mf1,mp_ep1,nu1,dv_ep1,Th_ep1] =
ep_maneuver(r_init,r_final,t_ep1,Isp_ep,Thr_ep,m_init,P_avail);
    end

    %Find what altitude increase can be achieved by EP in time period
    r_ep_final = sqrt(pos_ep1(1,end)^2+pos_ep1(2,end)^2);
    r_ep       = r_ep_final - r_init;
    r_ch_seg   = ((r_final - r_init) - (r_ep_final - r_init)) + r_init;

    [pos_ch1,vel_ch1,time_ch1,mf_ch1,dv_ch1,Th_ch1] =
chem_maneuver(r_ep_final,r_ch_seg,Isp_ch,m_init,nu1,t_const,imp);

    r_init_ep  = sqrt(pos_ch1(1,end)^2+pos_ch1(2,end)^2);
    m_init_ep  = m_init - mf_ch1;
    time_ep    = t_const - t_ep1 - time_ch1*hr2sec;

    while abs(r_init_ep - r_final) < 0.2
        r_ch_seg   = r_ch_seg - 1;
        [pos_ch1,vel_ch1,time_ch1,mf_ch1,dv_ch1,Th_ch1] =
chem_maneuver(r_ep_final,r_ch_seg,Isp_ch,m_init,nu1,t_const,imp);
        r_init_ep  = sqrt(pos_ch1(1,end)^2+pos_ch1(2,end)^2);
        m_init_ep  = m_init - mf_ch1;
        time_ep    = t_const - t_ep1 - time_ch1*hr2sec - 90; %Prevent
going over the time constraint.
    end
    [pos_ep2,vel_ep2,time_ep2,mf2,mp_ep2,nu2,dv_ep2,Th_ep2] =
ep_maneuver(r_init_ep,r_final,time_ep,Isp_ep,Thr_ep,m_init_ep,P_avail);
    time_check(kk) = time_ep2(end);
    r_ep_check(kk) = sqrt(pos_ep2(1,end)^2+pos_ep2(2,end)^2);
    while r_ep_check(kk) < r_final - 1
        r_ch_seg = r_ch_seg + 0.5;
        [pos_ch1,vel_ch1,time_ch1,mf_ch1,dv_ch1,Th_ch1] =
chem_maneuver(r_ep_final,r_ch_seg,Isp_ch,m_init,nu1,t_const,imp);
        r_init_ep  = sqrt(pos_ch1(1,end)^2+pos_ch1(2,end)^2);
        m_init_ep  = m_init - mf_ch1;
        time_ep    = t_const - t_ep1 - time_ch1*hr2sec - 90; %Prevent
going over the time constraint.
        [pos_ep2,vel_ep2,time_ep2,mf2,mp_ep2,nu2,dv_ep2,Th_ep2] =
ep_maneuver(r_init_ep,r_final,time_ep,Isp_ep,Thr_ep,m_init_ep,P_avail);
        time_check(kk) = time_ep2(end);
        r_ep_check(kk) = sqrt(pos_ep2(1,end)^2+pos_ep2(2,end)^2);
    end
    while time_ep2(end) < time_ep - 2*90
        r_ch_seg   = r_ch_seg - 0.5;
        [pos_ch1,vel_ch1,time_ch1,mf_ch1,dv_ch1,Th_ch1] =
chem_maneuver(r_ep_final,r_ch_seg,Isp_ch,m_init,nu1,t_const,imp);
        r_init_ep  = sqrt(pos_ch1(1,end)^2+pos_ch1(2,end)^2);
        m_init_ep  = m_init - mf_ch1;
        time_ep    = t_const - t_ep1 - time_ch1*hr2sec - 90;
```

```
        [pos_ep2,vel_ep2,time_ep2,mf2,mp_ep2,nu2,dv_ep2,Th_ep2] =
ep_maneuver(r_init_ep,r_final,time_ep,Isp_ep,Thr_ep,m_init_ep,P_avail);
    end
    if kk == 1
        rx_ep1a    = pos_ep1(1,:);
        ry_ep1a    = pos_ep1(2,:);
        rx_ep2a    = pos_ep2(1,:);
        ry_ep2a    = pos_ep2(2,:);
        rx_ch1a    = pos_ch1(1,:);
        ry_ch1a    = pos_ch1(2,:);
    end
    rx_ep1          = pos_ep1(1,:);
    ry_ep1          = pos_ep1(2,:);
    rx_ep2          = pos_ep2(1,:);
    ry_ep2          = pos_ep2(2,:);
    rx_ch1          = pos_ch1(1,:);
    ry_ch1          = pos_ch1(2,:);
    Ave_th_ece2(kk) = (Th_ep2*time_ep2(end) +
Th_ch1*time_ch1*hr2sec+Th_ep1*time_ep1(end))/(time_ep2(end) +
time_ep1(end) + time_ch1*hr2sec);
    r_ece2(kk)      = sqrt(pos_ep2(1,end)^2+pos_ep2(2,end)^2);
    mp_ece2(kk)     = mf_ch1 + mp_ep1 + mp_ep2;
    dv_ece2(kk)     = dv_ch1*km2m + dv_ep1(end) + dv_ep2(end);
    t_ece2(kk)      = time_ch1 + time_ep1(end)*sec2hr +
time_ep2(end)*sec2hr;
    kk              = kk + 1;
end
figure(10)
subplot(1,2,1)
plot(rx_ep1,ry_ep1,'--',rx_ch1,ry_ch1,'-x',rx_ep2,ry_ep2,'--
',rx_init,ry_init,rx_final,ry_final)
title('Electric - Chemical - Elecric Profile')
legend('1st Electric Segment','Chemical Segment','2nd Electric
Segment','Initial Orbit','Final Orbit')
subplot(1,2,2)
plot(rx_ep1a,ry_ep1a,'--',rx_ch1a,ry_ch1a,'-x',rx_ep2a,ry_ep2a,'--
',rx_init,ry_init,rx_final,ry_final)
title('Electric - Chemical - Elecric Profile')
legend('1st Electric Segment','Chemical Segment','2nd Electric
Segment','Initial Orbit','Final Orbit')
%***********************************************************
%DEVELOPING ANOTHER THRUST SCHEME, 28 Sep 10
%***********Chemical/Electric Propulsion System**************
%Initial Electric for x% of total time, chemical, electric for
remainder of time constraint.
kk = 1;
for i = min_ep:step_ep:max_ep
    %Calculate amount of time EP can thrust for, from 0 - 98% of time
    t_ep1       = i*t_const;
    r_init2     = (r_final - r_init)/2 + r_init;
    [pos_ep1,vel_ep1,time_ep1,mf1,mp_ep1,nu1,dv_ep1,Th_ep1] =
ep_maneuver(r_init2,r_final,t_ep1,Isp_ep,Thr_ep,m_init,P_avail);

    %Find what altitude increase can be achieved by EP in time period
    r_ep_final = sqrt(pos_ep1(1,end)^2+pos_ep1(2,end)^2);
    r_ep       = r_ep_final - r_init2;
```

```
    r_ep_final = r_init + r_ep;
    r_ch_seg   = ((r_final - r_init) - (r_ep_final - r_init)) + r_init;
    nu         = 0;

    [pos_ch1,vel_ch1,time_ch1,mf_ch1,dv_ch1,Th_ch1] =
chem_maneuver(r_init,r_ch_seg,Isp_ch,m_init,nu,t_const,imp);

    r_init_ep  = sqrt(pos_ch1(1,end)^2+pos_ch1(2,end)^2);
    m_init_ep  = m_init - mf_ch1;
    %time_ep    = t_const - time_ch1*hr2sec;

    if abs(r_init_ep - r_final) < 0.2
        Ave_th_ce(kk) =
(Th_ch1*time_ch1*hr2sec+Th_ep1*time_ep1(end))/(time_ep1(end) +
time_ch1*hr2sec);
        r_ce(kk)  = sqrt(pos_ch1(1,end)^2+pos_ch1(2,end)^2);
        mp_ce(kk) = mf_ch1;
        dv_ce(kk) = dv_ch1*km2m;
        t_ce(kk)  = time_ch1;
        kk        = kk + 1;
    else
        [pos_ep1,vel_ep1,time_ep1,mf1,mp_ep1,nu1,dv_ep1,Th_ep1] =
ep_maneuver(r_init_ep,r_final,t_ep1,Isp_ep,Thr_ep,m_init_ep,P_avail);
        while time_ep1(end) < t_ep1
            r_init_ep = r_init_ep - 1;
            [pos_ch1,vel_ch1,time_ch1,mf_ch1,dv_ch1,Th_ch1] =
chem_maneuver(r_init,r_init_ep,Isp_ch,m_init,nu,t_const,imp);
            r_init_ep  = sqrt(pos_ch1(1,end)^2+pos_ch1(2,end)^2);
            m_init_ep  = m_init - mf_ch1;
            [pos_ep1,vel_ep1,time_ep1,mf1,mp_ep1,nu1,dv_ep1,Th_ep1] =
ep_maneuver(r_init_ep,r_final,t_ep1,Isp_ep,Thr_ep,m_init_ep,P_avail);
        end
        rx_ep1        = pos_ep1(1,:);
        ry_ep1        = pos_ep1(2,:);
        rx_ch1        = pos_ch1(1,:);
        ry_ch1        = pos_ch1(2,:);
        Ave_th_ce(kk) =
(Th_ch1*time_ch1*hr2sec+Th_ep1*time_ep1(end))/(time_ep1(end) +
time_ch1*hr2sec);
        r_ce(kk)  = sqrt(pos_ep1(1,end)^2+pos_ep1(2,end)^2);
        mp_ce(kk) = mf_ch1 + mp_ep1;
        dv_ce(kk) = dv_ch1*km2m + dv_ep1(end);
        t_ce(kk)  = time_ch1 + time_ep1(end)*sec2hr;
        kk        = kk + 1;
    end
end
figure(11)
plot(rx_ep1,ry_ep1,'--',rx_ch1,ry_ch1,'-
x',rx_init,ry_init,rx_final,ry_final)
title('Chemical - Elecric Profile')
legend('Electric Segment','Chemical Segment','Initial Orbit','Final
Orbit')
ece_var  = [ mp_ece;  dv_ece;  t_ece;  Ave_th_ece;  r_ece];
ece2_var = [mp_ece2; dv_ece2; t_ece2; Ave_th_ece2; r_ece2];
cec_var  = [ mp_cec;  dv_cec;  t_cec;  Ave_th_cec;  r_cec];
ce_var   = [  mp_ce;   dv_ce;   t_ce;   Ave_th_ce;   r_ce];
```

```
ec_var   = [  mp_ec;   dv_ec;   t_ec;   Ave_th_ec;   r_ec];
```

Appendix C: Mission 3 MATLAB Code

The primary file used to model the variety of thrust duration for the plane change maneuver is the Miss3_orb_gravity.m file. This file is all inclusive and does not access any other files. The initial file developed to calculate the impulsive burn maneuvers was Run_M3.m. This file accesses planechange_1burn.m and planechange.m. The order the scripts are listed in this file are:

1) Miss3_orb_gravity.m

2) Run_M3.m

3) planechange_1burn.m

4) planechange.m

```
%This program calculates a plane change given an initial orbital
elements and propellant system properties
%Written by Tiffany Rexius 2010 - 2011
clear; clc;
%**********************Constants****************************
mu      = 3.986e5;  %[km^3/s^2]
g0      = 9.81;     %[m/s^2]
r_earth = 6378;     %[km]
hr2sec  = 60*60;    %[sec/hr]
sec2hr  = 1/hr2sec; %[hr/sec]
deg2rad = pi/180;   %[rad/deg]
rad2deg = 180/pi;   %[deg/rad]
km2m    = 1000;     %[m/km]
m2km    = 1/1000;   %[km/m]
%**********************************************************
%****************Set Initial Orbital Elements**************
ecc      = 0;   %[]    Initial Eccentricity
RAAN     = 0;   %[rad] Initial Right Ascension of the Ascending Node
arg0per  = 0;   %[rad] Initial Argument of Perigee
inc      = 0;   %[rad] Initial Inclination

ecc2     = 0;         %[]    Final Eccentricity
RAAN2    = 0;         %[rad] Final Right Ascension of the Ascending
Node
arg0per2 = 0;         %[rad] Final Argument of Perigee
inc_f    = 19*pi/180; %[rad] Final Inclination
inc1     = inc;       %[rad] Initial Inclination
```

```
nu       = 0;         %[rad] True Anomaly

p        = 6878;                 %[km]    Parameter
r        = p/(1+ecc2*cos(nu));   %[km]    Radius
P_init   = 2*pi*sqrt(r^3/mu);    %[sec]   Initial period of the orbit
w_init   = 2*pi/P_init;          %[rad/s] Initial orbital rate

dv_track = 0;                    %[km/s] Initial delta v
mp_tot   = 0;                    %[km/s] Initial propellant usage

step     = 1571;                 %[]    Number of steps per orbit
nu_step  = 2*pi/step;            %[rad] True anomaly step
t_step   = nu_step/w_init;       %[sec] Time step
ll       = 1;                    %[]    Count

%*********************Spacecraft Properties********************
m_prop = 80;                     %[kg]  Available Propellant
m_init = 180;                    %[kg]  Initial Spacecraft Mass
mass   = m_init;                 %[kg]  Spacecraft Mass
Th_ep  = 0.11;                   %[N]   Electric Propulsion Thrust
Isp_ep = 600;                    %[sec] Electric Propulsion Isp

%***********************State Vector***************************
r_pqw  = [r*cos(nu) r*sin(nu) 0];                 %[km]    Position
Vector
v_pqw  = sqrt(mu/p)*[-sin(nu) (ecc2+cos(nu)) 0]; %[km/s]  Velocity
Vector
r_pqw2 = r_pqw;                    %[km]    Transfer Orbit Position
Vector
v_pqw2 = v_pqw;                    %[km/s]  Transfer Orbit Velocity
Vector

v_ijk  = v_pqw*R313(RAAN,pi-inc,arg0per);  %[km/s] Velocity Vector in
GERF
r_ijk  = r_pqw*R313(RAAN,pi-inc,arg0per);  %[km]   Position Vector in
GERF
v_ijk2 = v_pqw2*R313(RAAN2,pi-inc_f,arg0per2);
r_ijk2 = r_pqw2*R313(RAAN2,pi-inc_f,arg0per2);

%*******************Transfer Orbit Properties******************
nu           = 0;         %[rad] True Anomaly
inc_tot      = 0;         %[rad] Inclination Change
nu_tol       = 0.0003;    %[]    Tolerance
t_count      = 0;         %[sec] Time tracker
nu_orb_track = 0;         %[]    Number of orbits completed
nu_track     = 0;         %[rad] True anomaly tracker
t_orb_track  = 0;         %[sec] Time in orbit tracker
kk           = 1;         %[]    Count

th_time      = 32*60;        %[sec] Thrust duration per orbit
t_max        = 200*24*3600;  %[sec] Time Limit

while t_count <= t_max %&& inc_tot <= inc_f %&& mass >= m_init - m_prop
    %Define gravity vector based on true anomoaly
```

```
    g_pqw   = [-mu/r^2*cos(nu) -mu/r^2*sin(nu) 0];  %[km/s]
    g_ijk   = g_pqw*R313(RAAN,pi-inc(kk),arg0per);  %[km/s]
    g_ijk2  = g_pqw*R313(RAAN2,pi-inc_f,arg0per2);  %[km/s]

    %Define Final spacecraft position
    v_pqw_final    = sqrt(mu/p)*[-sin(nu) (ecc+cos(nu)) 0];     %[km/s]
    r_pqw_final    = [r*cos(nu) r*sin(nu) 0];                   %[km]
    r_ijk2(kk,:)   = r_pqw_final*R313(RAAN2,pi-inc_f,arg0per2); %[km]
    v_ijk2(kk,:)   = v_pqw_final*R313(RAAN2,pi-inc_f,arg0per2); %[km/s]

    %Define dv direction
    dv_diff        = v_ijk2(kk,:) - v_ijk(kk,:);         %[km/s]
    dv_vec         = dv_diff/mag(dv_diff);               %[km/s]

    %Calulate Final mass
    mf             = mass - t_step*(Th_ep/(Isp_ep*g0));   %[kg]
% %***************************Thrust Continuously********************
%     dv             = Th_ep*t_step/mass*m2km;
%     v_ijk(kk+1,:)  = v_ijk(kk,:) + t_step*g_ijk + dv*dv_vec;
%     r_ijk(kk+1,:)  = r_ijk(kk,:) + t_step*v_ijk(kk,:);
%     h_vec(kk+1,:)  = cross(r_ijk(kk+1,:),v_ijk(kk+1,:));
%     inc(kk+1)      = acos(abs(h_vec(kk+1,3))/mag(h_vec(kk+1,:)));
%
%*******************************************************************
% %***********Thrust twice beginning at the crossover points********
%     if kk <= th_time*w_init/nu_step     %Thrusting
%         dv             = Th_ep*t_step/mass*m2km;
%         v_ijk(kk+1,:)  = v_ijk(kk,:) + t_step*g_ijk + dv*dv_vec;
%         r_ijk(kk+1,:)  = r_ijk(kk,:) + t_step*v_ijk(kk,:);
%         h_vec(kk+1,:)  = cross(r_ijk(kk+1,:),v_ijk(kk+1,:));
%         inc(kk+1)      = acos(abs(h_vec(kk+1,3))/mag(h_vec(kk+1,:)));
% %     elseif t_orb_track >= P_init/2 && t_orb_track <= P_init/2 +
th_time
% %         dv             = Th_ep*t_step/mass*m2km;
% %         v_ijk(kk+1,:)  = v_ijk(kk,:) + t_step*g_ijk + dv*dv_vec;
% %         r_ijk(kk+1,:)  = r_ijk(kk,:) + t_step*v_ijk(kk,:);
% %         h_vec(kk+1,:)  = cross(r_ijk(kk+1,:),v_ijk(kk+1,:));
% %         inc(kk+1)      =
acos(abs(h_vec(kk+1,3))/mag(h_vec(kk+1,:)));
%     else
%         dv             = 0;
%         v_ijk(kk+1,:)  = v_ijk(kk,:) + t_step*g_ijk;
%         r_ijk(kk+1,:)  = r_ijk(kk,:) + t_step*v_ijk(kk,:);
%         h_vec(kk+1,:)  = cross(r_ijk(kk,:),v_ijk(kk,:));
%         inc(kk+1)      = acos(abs(h_vec(kk+1,3))/mag(h_vec(kk+1,:)));
%     end
% ******************************************************************
% *********Thrust twice centered on the crossover points************
    if kk <= th_time/2*w_init/nu_step   %Thrusting
        dv             = -Isp_ep*g0*log(mf/mass)*m2km;%
        dv_check       = Th_ep*t_step/mass*m2km;
        v_ijk(kk+1,:)  = v_ijk(kk,:) + t_step*g_ijk + dv*dv_vec;
        r_ijk(kk+1,:)  = r_ijk(kk,:) + t_step*v_ijk(kk,:);
        h_vec(kk+1,:)  = cross(r_ijk(kk+1,:),v_ijk(kk+1,:));
        inc(kk+1)      = acos(abs(h_vec(kk+1,3))/mag(h_vec(kk+1,:)));
```

```
    elseif t_orb_track >= P_init/2 - th_time/2 && t_orb_track <=
P_init/2 + th_time/2
        dv              = -Isp_ep*g0*log(mf/mass)*m2km;
        v_ijk(kk+1,:)   = v_ijk(kk,:) + t_step*g_ijk + dv*dv_vec;
        r_ijk(kk+1,:)   = r_ijk(kk,:) + t_step*v_ijk(kk,:);
        h_vec(kk+1,:)   = cross(r_ijk(kk+1,:),v_ijk(kk+1,:));
        inc(kk+1)       = acos(abs(h_vec(kk+1,3))/mag(h_vec(kk+1,:)));
    elseif t_orb_track >= P_init - th_time/2
        dv              = -Isp_ep*g0*log(mf/mass)*m2km;
        v_ijk(kk+1,:)   = v_ijk(kk,:) + t_step*g_ijk + dv*dv_vec;
        r_ijk(kk+1,:)   = r_ijk(kk,:) + t_step*v_ijk(kk,:);
        h_vec(kk+1,:)   = cross(r_ijk(kk+1,:),v_ijk(kk+1,:));
        inc(kk+1)       = acos(abs(h_vec(kk+1,3))/mag(h_vec(kk+1,:)));
    else
        dv              = 0;
        v_ijk(kk+1,:)   = v_ijk(kk,:) + t_step*g_ijk;
        r_ijk(kk+1,:)   = r_ijk(kk,:) + t_step*v_ijk(kk,:);
        h_vec(kk+1,:)   = cross(r_ijk(kk,:),v_ijk(kk,:));
        inc(kk+1)       = acos(abs(h_vec(kk+1,3))/mag(h_vec(kk+1,:)));
    end
%*******************************************************************
%Update position in transfer orbit
    r_pqw_check         = [r*cos(nu+nu_step) r*sin(nu+nu_step) 0];
    r_ijk_check(kk+1,:) = r_pqw_check*R313(RAAN,pi-inc(kk+1),arg0per);
    v_pqw_check         = sqrt(mu/p)*[-sin(nu+nu_step)
(ecc+cos(nu+nu_step)) 0];
    v_ijk_check(kk+1,:) = v_pqw_check*R313(RAAN,pi-inc(kk+1),arg0per);
    h_vec_check         =
cross(r_ijk_check(kk+1,:),v_ijk_check(kk+1,:));
    inc_check(kk+1)     = pi - acos(h_vec_check(3)/mag(h_vec_check));

    %Update state vector based on calculated inclination. This causes
the crossover point not to shift.

    r_ijk(kk+1,:)    = r_ijk_check(kk+1,:);
    v_ijk(kk+1,:)    = v_ijk_check(kk+1,:);

    %******************Calculate and track dv and mass***************
    dv_vec2(kk,:)  = dv*dv_vec;
    dv2_mag(kk)    = mag(dv_vec2(kk,:));
    dv_track       = dv_track + dv;
    mp             = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
    mp2 = mass - mf;
    mp_tot         = mp_tot + mp;                        %[kg]
    mass           = mass - mp;                          %[kg]

    %Track nu each orbit
    nu_orbit     = t_step*w_init;
    if kk > 1
        nu_orb_track(kk) = nu_orb_track(kk-1) + nu_orbit;
    end

    t_orb_track  = nu_orb_track(kk)/w_init;
    nu       = nu + nu_step;
    nu_track = nu_track + nu_step;
```

```
    t_count = nu_track/w_init;
    inc_tot = inc_check(kk);
    inc2    = inc';
    n_orbit = t_count/5677.5;
    nu_kk   = abs(n_orbit - round(n_orbit));
    if t_count> 90*3600*24
        inc_90day       = inc(end);
        dv_90day       = dv_track;
        mp_90day        = mp_tot;
        t_90day         = t_count/3600/24;
    end
    if abs(t_count - 2*24*3600) < 2;
        inc_2day       = inc(end);
        dv_2day        = dv_track;
        mp_2day        = mp_tot;
        t_2day         = t_count/3600/24;
%         inc_2deg       = inc(end);
%         dv_2deg        = dv_track;
%         mp_2deg        = mp_tot;
%         t_2deg         = t_count/3600/24;
     elseif abs(t_count - 10*24*3600) < 2;
        inc_10day      = inc(end);
        dv_10day       = dv_track;
        mp_10day       = mp_tot;
        t_10day        = t_count/3600/24;
%         inc_4deg       = inc(end);
%         dv_4deg        = dv_track;
%         mp_4deg        = mp_tot;
%         t_4deg         = t_count/3600/24;
    elseif abs(t_count - 20*24*3600) < 2; %inc(kk) >= 0.1047 &&
inc(kk)<=0.1048
        inc_20day      = inc(end);
        dv_20day       = dv_track;
        mp_20day       = mp_tot;
        t_20day        = t_count/3600/24;
%         inc_6deg       = inc(end);
%         dv_6deg        = dv_track;
%         mp_6deg        = mp_tot;
%         t_6deg         = t_count/3600/24;
    elseif abs(t_count - 30*24*3600) < 2; %inc(kk) >= 0.1396 &&
inc(kk)<= 0.1397
        inc_30days = inc(end);
        dv_30days  = dv_track;
        mp_30days  = mp_tot;
        t_30days   = t_count/3600/24;
%         inc_30days = inc(end);
%         dv_30days  = dv_track;
%         mp_30days  = mp_tot;
%         t_8deg     = t_count/3600/24;
    elseif abs(t_count - 40*24*3600) < 2; %inc(kk) >= 0.1745 &&
inc(kk)<= 0.1746
        inc_40days = inc(end);
        dv_40days  = dv_track;
        mp_40days  = mp_tot;
        t_40days   = t_count/3600/24;
%         inc_40days = inc(end);
```

```
%          dv_40days  = dv_track;
%          mp_40days  = mp_tot;
%          t_10deg    = t_count/3600/24;
    elseif abs(t_count - 50*24*3600) < 2; %inc(kk) >= 0.2094 &&
inc(kk)<= 0.2095
        inc_50days  = inc(end);
        dv_50days   = dv_track;
        mp_50days   = mp_tot;
        t_50days    = t_count/3600/24;
%          inc_12deg  = inc(end);
%          dv_12deg   = dv_track;
%          mp_12deg   = mp_tot;
%          t_12deg    = t_count/3600/24;
    elseif abs(t_count - 60*24*3600) < 2; %inc(kk) >= 0.2443 &&
inc(kk)<= 0.2444
        inc_60days = inc(end);
        dv_60days  = dv_track;
        mp_60days  = mp_tot;
        t_60days   = t_count/3600/24;
%          inc_14deg = inc(end);
%          dv_14deg  = dv_track;
%          mp_14deg  = mp_tot;
%          t_14deg   = t_count/3600/24;
    elseif abs(t_count - 70*24*3600) < 2; %inc(kk) >= 0.2618 &&
inc(kk)<= 0.2619
        inc_70days  = inc(end);
        dv_70days   = dv_track;
        mp_70days   = mp_tot;
        t_70days    = t_count/3600/24;
%          inc_15deg = inc(end);
%          dv_15deg  = dv_track;
%          mp_15deg  = mp_tot;
%          t_15deg   = t_count/3600/24;
    elseif abs(t_count - 80*24*3600) < 2;
        inc_80days  = inc(end);
        dv_80days   = dv_track;
        mp_80days   = mp_tot;
        t_80days    = t_count/3600/24;
    elseif abs(t_count - 90*24*3600) < 2;
        inc_90days  = inc(end);
        dv_90days   = dv_track;
        mp_90days   = mp_tot;
        t_90days    = t_count/3600/24;
   elseif abs(t_count - 100*24*3600) < 2;
        inc_100days  = inc(end);
        dv_100days   = dv_track;
        mp_100days   = mp_tot;
        t_100days    = t_count/3600/24;
   elseif abs(t_count - 110*24*3600) < 2;
        inc_110days  = inc(end);
        dv_110days   = dv_track;
        mp_110days   = mp_tot;
        t_110days    = t_count/3600/24;
   elseif abs(t_count - 120*24*3600) < 2;
        inc_120days  = inc(end);
        dv_120days   = dv_track;
```

```
        mp_120days   = mp_tot;
        t_120days    = t_count/3600/24;
    end
    kk    = kk + 1;
    if kk == step + 1
        dv2_mag_track = dv2_mag';
        r_ijk(1,:) = r_ijk(kk-1,:);
        v_ijk(1,:) = v_ijk(kk-1,:);
        inc(1)     = inc(kk-1);
        kk               = 1;
        nu_orb_track(kk) = 0;
        nu               = nu - 2*pi;
    end
end
plot3(r_ijk_check(:,1),r_ijk_check(:,2),r_ijk_check(:,3),'-
*',r_ijk2(:,1),r_ijk2(:,2),r_ijk2(:,3));
```

Run_M3.m file

```
%This program runs Reference Mission 3, which is a 15 degree plane
change
%in 90 days.

%Currently this is modeled using equations from "Fundamentals of
%Astrodynamics" by Bate, Mueller, White.

clc;
clear;
%***************Vehicle Information************************
m_init      = 180;  %[kg]
m_prop      = 80;   %[kg]
P_avail     = 1000; %[W]
Isp_ep      = 600;  %[s]
Thrust_ep   = 0.11; %[N]
Isp_chem    = 235;  %[s]
imp         = 20*60; %[sec] This represents a 20 min burn arc
%*********************************************************
%********************Constants****************************
mu      = 3.986e5;  %[km^3/s^2]
g0      = 9.81;     %[m/s^2]
r_earth = 6378;     %[km]
hr2sec  = 60*60;    %[sec/hr]
sec2hr  = 1/hr2sec; %[hr/sec]
deg2rad = pi/180;   %[rad/deg]
rad2deg = 180/pi;   %[deg/rad]
km2m    = 1000;     %[m/km]
m2km    = 1/1000;   %[km/m]
%*********************************************************
%*********************Mission 1****************************
t_const = 90*24*hr2sec;          %[sec] Time Limit
inc1    = 0*deg2rad;             %[rad] Plane Change Angle
inc2    = 15*deg2rad;            %[rad] Plane Change Angle
r_init  = 500 + r_earth;         %[km]
```

```
ecc     = 0;                        %[--] Circular orbit
RAAN    = 0;                        %[rad] Right Ascension of the
Ascending Node
arg0per = 0;                        %[rad] Argument of Periapsis
p       = r_init;                   %[km] Periapsis
a_init  = r_init;                   %[km] Circular orbit
P       = 2*pi*sqrt(a_init^3/mu);   %[sec]
w_init  = 2*pi/P;                   %[rad/s]
V       = sqrt(mu/r_init);          %[km/s]
dv      = 2*V*sin(inc2/2);          %[km/s]

[TP2,TP4,TP6] =
planechange_1burn(m_init,Isp_ep,Thrust_ep,p,ecc,inc1,inc2,RAAN,arg0per,
t_const,m_prop);

[Th_prof1, Th_prof2, Th_prof3, Th_prof4, Th_prof5, Th_prof6] =
planechange(m_init,Isp_ep,Thrust_ep,Isp_chem,p,ecc,inc1,inc2,RAAN,arg0p
er,t_const,m_prop);

inc_ep1 = Th_prof1(1);
t_ep1   = Th_prof1(2);
mp_ep1  = Th_prof1(3);
dv_ep1  = Th_prof1(4);

inc_ep2 = Th_prof2(1);
t_ep2   = Th_prof2(2);
mp_ep2  = Th_prof2(3);
dv_ep2  = Th_prof2(4);

inc_ep3 = Th_prof3(1);
t_ep3   = Th_prof3(2);
mp_ep3  = Th_prof3(3);
dv_ep3  = Th_prof3(4);

inc_ep4 = Th_prof4(1);
t_ep4   = Th_prof4(2);
mp_ep4  = Th_prof4(3);
dv_ep4  = Th_prof4(4);

inc_ep5 = Th_prof5(1);
t_ep5   = Th_prof5(2);
mp_ep5  = Th_prof5(3);
dv_ep5  = Th_prof5(4);

inc_ep6 = Th_prof6(1);
t_ep6   = Th_prof6(2);
mp_ep6  = Th_prof6(3);
dv_ep6  = Th_prof6(4);

mp_chem      = m_init - m_init*exp(-dv*km2m/Isp_chem/g0);  %[kg]
mass_ep      = m_init - m_init*exp(-dv*km2m/Isp_ep/g0);    %[kg]

burn_time = imp*2;
Th_ch     = m_init*dv*km2m/(burn_time); %[N]
```

```
Th_ep     = m_init*dv*km2m/(t_const/2); %[N]

fprintf('*********Results for Mission 3***********\n\n')
fprintf('*************Chemical System*************\n')
fprintf('   Delta V (m/s)              = %6.2f\n',dv*km2m)
fprintf('   Chem Propellant Mass (kg) = %6.2f\n',mp_chem)
fprintf('   Average Thrust Chem (N)   = %6.2f\n',Th_ch)
fprintf('   EP Propellant Mass (kg)   = %6.2f\n',mass_ep)
fprintf('*****************************************\n')
fprintf('*************Electric System************\n')
fprintf('**Thrust Profile 1: 2 - 15 min Burns*****\n')
fprintf('   Delta V (m/s)              = %6.2f\n',dv_ep1*km2m)
fprintf('   Elec Propellant Mass (kg) = %6.2f\n',mp_ep1)
fprintf('   Inclination Change (deg)  = %6.2f\n',inc_ep1*rad2deg)
fprintf('   Trip Time (days)          = %6.2f\n',t_ep1*sec2hr/24)
fprintf('*****************************************\n')
fprintf('**Thrust Profile 2: 1 - 15 min Burn******\n')
fprintf('   Delta V (m/s)              = %6.2f\n',dv_ep2*km2m)
fprintf('   Elec Propellant Mass (kg) = %6.2f\n',mp_ep2)
fprintf('   Inclination Change (deg)  = %6.2f\n',inc_ep2*rad2deg)
fprintf('   Trip Time (days)          = %6.2f\n',t_ep2*sec2hr/24)
fprintf('*****************************************\n')
fprintf('**Thrust Profile 4: 15 - 1 min Burns******\n')
fprintf('   Delta V (m/s)              = %6.2f\n',dv_ep4*km2m)
fprintf('   Elec Propellant Mass (kg) = %6.2f\n',mp_ep4)
fprintf('   Inclination Change (deg)  = %6.2f\n',inc_ep4*rad2deg)
fprintf('   Trip Time (days)          = %6.2f\n',t_ep4*sec2hr/24)
fprintf('*****************************************\n')
fprintf('*Thrust Profile 5: Continous 1-min Burns*\n')
fprintf('   Delta V (m/s)              = %6.2f\n',dv_ep5*km2m)
fprintf('   Elec Propellant Mass (kg) = %6.2f\n',mp_ep5)
fprintf('   Inclination Change (deg)  = %6.2f\n',inc_ep5*rad2deg)
fprintf('   Trip Time (days)          = %6.2f\n',t_ep5*sec2hr/24)
fprintf('*****************************************\n')
fprintf('**Thrust Profile 6: 15 - 1 min Burns nu=0**\n')
fprintf('   Delta V (m/s)              = %6.2f\n',dv_ep6*km2m)
fprintf('   Elec Propellant Mass (kg) = %6.2f\n',mp_ep6)
fprintf('   Inclination Change (deg)  = %6.2f\n',inc_ep6*rad2deg)
fprintf('   Trip Time (days)          = %6.2f\n',t_ep6*sec2hr/24)
fprintf('*****************************************\n')
fprintf('*************Combined System*************\n')
fprintf('**Thrust Profile 3: 2 - 15 min Burns*****\n')
fprintf('*****Followed by Chemical Manuever*******\n')
fprintf('   Delta V (m/s)              = %6.2f\n',dv_ep3*km2m)
fprintf('   Elec Propellant Mass (kg) = %6.2f\n',mp_ep3)
fprintf('   Inclination Change (deg)  = %6.2f\n',inc_ep3*rad2deg)
fprintf('   Trip Time (days)          = %6.2f\n',t_ep3*sec2hr/24)
```

Planechange_1burn.m file

```
function [Th_prof2_track,Th_prof4_track,Th_prof6_track] =
planechange_1burn(m_init,Isp_ep,Thrust_ep,p,ecc,inc1,inc2,RAAN,arg0per,
t_const,m_prop)
```

```
%*********************Constants*****************************
mu      = 3.986e5;  %[km^3/s^2]
g0      = 9.81;     %[m/s^2]
r_earth = 6378;     %[km]
hr2sec  = 60*60;    %[sec/hr]
sec2hr  = 1/hr2sec; %[hr/sec]
deg2rad = pi/180;   %[rad/deg]
rad2deg = 180/pi;   %[deg/rad]
km2m    = 1000;     %[m/km]
m2km    = 1/1000;   %[km/m]
%*************************************************************
nu     = 0;
r      = p/(1+ecc*cos(nu));
P_init = 2*pi*sqrt(r^3/mu);
w_init = 2*pi/P_init;
kk     = 1;
pp     = 1;
t_min_range = 4/60*hr2sec;
t_max_range = 25/60*hr2sec;

%This saves the orbit in the Geocentric Equitorial Frame for one
rotation for the initial and final orbit.  This is used to determine
the necessary thrust vector.
for nu = 0:0.1:2*pi+0.1
    [r_ijk v_ijk] = pqw2ijk(p,ecc,inc1,RAAN,arg0per,nu);
    r_ijk1(kk,:)  = r_ijk;
    v_ijk1(kk,:)  = v_ijk;

    [r_ijk v_ijk] = pqw2ijk(p,ecc,inc2,RAAN,arg0per,nu);
    r_ijk2(kk,:)  = r_ijk;
    v_ijk2(kk,:)  = v_ijk;

    if nu <= pi+0.1
        r_ijk2a(kk,:)  = r_ijk;
        v_ijk2a(kk,:)  = v_ijk;
    else
        r_ijk2b(pp,:)  = r_ijk;
        v_ijk2b(pp,:)  = v_ijk;
        pp = pp + 1;
    end
    kk = kk+1;
end

h2   = cross(r_ijk2(1,:),v_ijk2(1,:));
inc2 = abs(pi- acos(h2(3)/mag(h2)));

%*********************************************************************
%**************Thrust Scheme 2: 1 - 15 min burn at 0****************
%*********************************************************************
%*********************Re-initialize Variables***********************
mass        = m_init;         %[kg]
t_max       = t_const;        %[sec]
jj          = 1;              %[--]
mm          = 1;              %[--]
```

```
time        = 0;               %[sec]
v_ijk3(1,:) = v_ijk1(1,:);     %[km/s]
r_ijk3(1,:) = r_ijk1(1,:);     %[km/s]
dv_track    = 0;               %[km/s]
inc         = 0;               %[rad]
mp_tot      = 0;               %[kg]
%*******************************************************************
for t_step = t_min_range:1/60*hr2sec:t_max_range
    while time < t_max && inc < inc2 && mass > m_init-m_prop
        dv              = Thrust_ep*t_step/mass*m2km;         %[km/s]
        dv_track        = dv_track + dv;                      %[km/s]
        mp              = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
        mp_tot          = mp_tot + mp;                        %[kg]
        mass            = mass - mp;                          %[kg]

        nu      = w_init*time;
        v_pqw2a = sqrt(mu/p)*[-sin(nu) (ecc+cos(nu)) 0];


        %Calulate updated thrust vector between transfer & final orbit
        [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2a,dv,t_step);

        %Calculate updated inclination between transfer & final orbit
%         h3  = cross(r_ijk3,v_ijk3);
%         inc = abs(pi - acos(h3(3)/mag(h3)));

        [RAAN, arg0perx, eccx,incx] = orbparam(r_ijk3,v_ijk3);
        inc = pi - incx;

        %Transfer from Geocentric Equatorial to Perifocal
        r_pqw3       = r_ijk3*R313(arg0per,pi-inc,RAAN);    %[km]
        v_pqw3       = v_ijk3*R313(arg0per,pi-inc,RAAN);    %[km/s]

        %Calculate orbit from 0 to 2pi after thrust
        for nu = 0:0.1:2*pi+0.1
            r_pqw(jj,:) = [mag(r_pqw3)*cos(nu) mag(r_pqw3)*sin(nu) 0];
            v_pqw(jj,:) = sqrt(mu/p)*[-sin(nu) (ecc+cos(nu)) 0];
            jj          = jj + 1;
        end
        jj            = 1;
        r_ijk3        = r_pqw*R313(RAAN,pi-inc,arg0per); %[km]
        v_ijk3        = v_pqw*R313(RAAN,pi-inc,arg0per); %[km/s]
        time          = time + P_init;                   %[sec]Update
time
    end
    Th_prof2_track(mm,:) = [inc, time, mp_tot,dv_track];
    orbit_param(mm,:)    = [RAAN, arg0per, eccx,inc];
    mm          = mm + 1;
    time        = 0;
    inc         = 0;
    mp_tot      = 0;
    dv_track    = 0;
    mass        = m_init;
    v_ijk3(1,:) = v_ijk1(1,:);    %[km/s]
    r_ijk3(1,:) = r_ijk1(1,:);    %[km/s]
```

```
end
%**********************************************************************
%***************Thrust Scheme 4: 15 - 1 min burns******************
%**********************************************************************
mass         = m_init;        %[kg]
t_step_min   = 1*60;          %[sec]
t_max        = t_const;       %[sec]
jj           = 1;             %[--]
mm           = 1;             %[--]
time         = 0;             %[sec]
v_ijk3(1,:) = v_ijk1(1,:);    %[km/s]
r_ijk3(1,:) = r_ijk1(1,:);    %[km/s]
v_pqw2f      = sqrt(mu/p)*[-sin(w_init*t_step_min)
(ecc+cos(w_init*t_step_min)) 0];
v_ijk2f      = v_pqw2f*R313(RAAN,pi-inc,arg0per);
dv_track     = 0;             %[km/s]
inc          = 0;             %[rad]
mp_tot       = 0;             %[kg]
%****************************************************************
for t_step = t_min_range:1/60*hr2sec:t_max_range
    while time < t_max && inc < inc2 && mass > m_init-m_prop
        for time2 = t_step_min:t_step_min:t_step
            dv         = Thrust_ep*t_step_min/mass*m2km;     %[km/s]
            dv_track   = dv_track + dv;                      %[km/s]
            mp         = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
            mp_tot     = mp_tot + mp;                        %[kg]
            mass       = mass - mp;                          %[kg]

            %Calulate updated thrust vector between transfer & final
orbit
            [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2f,-
dv,time2);

            %Calculate updated inclination between transfer & final
orbit
            h3  = cross(r_ijk3,v_ijk3);                     %[km^2/s]
            inc = abs(pi - acos(h3(3)/mag(h3)));            %[rad]
            %Transfer from Geocentric Equatorial to Perifocal
            r_pqw3a      = r_ijk3*R313(arg0per,pi-inc,RAAN);     %[km]
            nua          = w_init*time2;                         %[rad]
            r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nua) mag(r_pqw3a)*sin(nua)
0];
            v_pqw3(jj,:) = sqrt(mu/mag(r_pqw3a))*[-sin(nua)
(ecc+cos(nua)) 0];
            r_ijk3     = r_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per); %[km]
            v_ijk3     = v_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per); %[km/s]

            %Calc final orbit velocity to get correct thrust vector
            v_pqw2f = sqrt(mu/p)*[-sin(nua) (ecc+cos(nua)) 0];
            v_ijk2f = v_pqw2f*R313(RAAN,pi-inc,arg0per);

            if time2 == t_step
                r_ijk3 = r_pqw3*R313(RAAN,pi-inc,arg0per);
                v_ijk3 = v_pqw3*R313(RAAN,pi-inc,arg0per);
```

```
            end
            jj = jj + 1;
        end
        jj = jj - 1;

        %Calculate orbit after thrust
        for nu = nua:0.1:(2*pi+nua)
            r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nu) mag(r_pqw3a)*sin(nu)
0];
            v_pqw3(jj,:) = sqrt(mu/p)*[-sin(nu) (ecc+cos(nu)) 0];
            jj           = jj + 1;
        end
        r_ijk3   = r_pqw3*R313(RAAN,pi-inc,arg0per);
        v_ijk3   = v_pqw3*R313(RAAN,pi-inc,arg0per);
        jj       = 1;
        time     = time + P_init;  %[sec] Update time
    end
    Th_prof4_track(mm,:) = [inc, time, mp_tot,dv_track];
    mm       = mm + 1;
    time     = 0;
    inc      = 0;
    mp_tot   = 0;
    dv_track = 0;
    mass     = m_init;
    v_ijk3(1,:) = v_ijk1(1,:);    %[km/s]
    r_ijk3(1,:) = r_ijk1(1,:);    %[km/s]
end

%*******************************************************************
%****************Thrust Scheme 6: 15 - 1 min burns******************
%**************Centered around crossover points*********************
%*******************************************************************
mass        = m_init;         %[kg]
t_step_min  = 0.5*60;         %[sec]
t_max       = t_const;        %[sec]
jj          = 1;              %[--]
nn          = 1;              %[--]
time        = 0;              %[sec]
v_ijk3(1,:) = v_ijk1(1,:);    %[km/s]
r_ijk3(1,:) = r_ijk1(1,:);    %[km/s]
dv_track    = 0;              %[km/s]
inc         = 0;              %[rad]
mp_tot      = 0;              %[kg]
%***********************************************************

for t_step = t_min_range:1/60*hr2sec:t_max_range
    while time < t_max && inc < inc2 && mass > m_init-m_prop
        for time2 = t_step_min:t_step_min:(t_step/2)
            dv        = Thrust_ep*t_step_min/mass*m2km;     %[km/s]
            dv_track  = dv_track + dv;                      %[km/s]
            mp        = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
            mp_tot    = mp_tot + mp;                        %[kg]
            mass      = mass - mp;                          %[kg]
```

```
            %Calulate updated thrust vector between transfer & final
orbit
            if time2 > 0
                [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2f,-
dv,time2);
            else
                [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2,-
dv,time2);
            end

%Calculate updated inclination between transfer & final orbit
            h3  = cross(r_ijk3,v_ijk3);                    %[km^2/s]
            inc = abs(pi - acos(h3(3)/mag(h3)));           %[rad]

            %Transfer from Geocentric Equatorial to Perifocal
            r_pqw3a      = r_ijk3*R313(arg0per,pi-inc,RAAN);    %[km]
            nua          = w_init*time2;                        %[rad]
            r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nua) mag(r_pqw3a)*sin(nua)
0];
            v_pqw3(jj,:) = sqrt(mu/mag(r_pqw3a))*[-sin(nua)
(ecc+cos(nua)) 0];
            r_ijk3    = r_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per); %[km]
            v_ijk3    = v_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per); %[km/s]

            %Calc final orbit velocity to get correct thrust vector
            v_pqw2f = sqrt(mu/p)*[-sin(nua) (ecc+cos(nua)) 0];
            v_ijk2f = v_pqw2f*R313(RAAN,pi-inc,arg0per);

            if time2 == t_step
                r_ijk3 = r_pqw3*R313(RAAN,pi-inc,arg0per);
                v_ijk3 = v_pqw3*R313(RAAN,pi-inc,arg0per);
            end
            jj = jj + 1;
        end
        jj = jj - 1;

        %Calculate orbit after thrust
        for nu = nua:0.01:(2*pi-nua)
            r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nu) mag(r_pqw3a)*sin(nu)
0];
            v_pqw3(jj,:) = sqrt(mu/p)*[-sin(nu) (ecc+cos(nu)) 0];
            jj = jj + 1;
        end
        nub    = nu;
        time2  = nub/w_init;
        r_ijk3 = r_pqw3*R313(RAAN,pi-inc,arg0per);
        v_ijk3 = v_pqw3*R313(RAAN,pi-inc,arg0per);

        for time2 = (time2):t_step_min:(time2+t_step/2)
            dv        = Thrust_ep*t_step_min/mass*m2km;     %[km/s]
            dv_track  = dv_track + dv;                      %[km/s]
            mp        = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
            mp_tot    = mp_tot + mp;                        %[kg]
            mass      = mass - mp;                          %[kg]
```

```
  %Calulate updated thrust vector between transfer & final orbit
            [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2f,dv,time2);

  %Calculate updated inclination between transfer & final orbit
            h3  = cross(r_ijk3,v_ijk3);                        %[km^2/s]
            inc = abs(pi - acos(h3(3)/mag(h3)));               %[rad]

            %Transfer from Geocentric Equatorial to Perifocal
            r_pqw3a      = r_ijk3*R313(arg0per,pi-inc,RAAN);    %[km]
            nua          = w_init*time2;                        %[rad]
            r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nua) mag(r_pqw3a)*sin(nua)
0];
            v_pqw3(jj,:) = sqrt(mu/mag(r_pqw3a))*[-sin(nua)
(ecc+cos(nua)) 0];
            r_ijk3    = r_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per); %[km]
            v_ijk3    = v_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per); %[km/s]

            %Calc final orbit velocity is to get correct thrust vector
            v_pqw2f = sqrt(mu/p)*[-sin(nub) (ecc+cos(nub)) 0];
            v_ijk2f = v_pqw2f*R313(RAAN,pi-inc,arg0per);

            if time2 == t_step
                r_ijk3 = r_pqw3*R313(RAAN,pi-inc,arg0per);
                v_ijk3 = v_pqw3*R313(RAAN,pi-inc,arg0per);
            end
            jj = jj + 1;
        end
        r_ijk3   = r_pqw3*R313(RAAN,pi-inc,arg0per);    %[km]
        v_ijk3   = v_pqw3*R313(RAAN,pi-inc,arg0per);    %[km]
        jj       = 1;                                   %[--]
        time     = time + P_init;  %[sec] Update time
    end
    Th_prof6_track(nn,:) = [inc, time, mp_tot,dv_track];
    nn       = nn + 1;
    time     = 0;
    inc      = 0;
    mp_tot   = 0;
    dv_track = 0;
    mass     = m_init;
    v_ijk3(1,:) = v_ijk1(1,:);    %[km/s]
    r_ijk3(1,:) = r_ijk1(1,:);    %[km/s]
end
t_plot  =
linspace(t_min_range,t_max_range,length(Th_prof6_track(:,1)));

figure(1)
subplot(3,1,1)
plot(t_plot/60,Th_prof2_track(:,1)*rad2deg,'x',t_plot/60,Th_prof4_track
(:,1)*rad2deg,'^',t_plot/60,Th_prof6_track(:,1)*rad2deg,'>')
legend('1 Burn at nu = 0','X incremental Burns','1 Burn centered at \nu
= 0')
ylabel('Inclination Change (deg)')
xlabel('Thrust Duration (min)')
subplot(3,1,2)
```

```
plot(t_plot/60,Th_prof2_track(:,3),'x',t_plot/60,Th_prof4_track(:,3),'^
',t_plot/60,Th_prof6_track(:,3),'>')
legend('1 Burn at nu = 0','X incremental Burns','1 Burn centered at \nu
= 0')
ylabel('Propellant Mass (kg))')
xlabel('Thrust Duration (min)')
subplot(3,1,3)
plot(Th_prof2_track(:,1)*rad2deg,Th_prof2_track(:,4)*km2m,'x',Th_prof4_
track(:,1)*rad2deg,Th_prof4_track(:,4)*km2m,'^',Th_prof6_track(:,1)*rad
2deg,Th_prof6_track(:,4)*km2m,'>')
legend('1 Burn at nu = 0','X incremental Burns','1 Burn centered at \nu
= 0')
xlabel('Inclination Change (deg)')
ylabel('Delta V (m/s)')
```

Planechange.m file

```
function [Th_prof1, Th_prof2, Th_prof3,Th_prof4, Th_prof5, Th_prof6] =
planechange(m_init,Isp_ep,Thrust_ep,Isp_chem,p,ecc,inc1,inc2,RAAN,arg0p
er,t_const,m_prop)

%*********************Constants*****************************
mu      = 3.986e5;  %[km^3/s^2]
g0      = 9.81;     %[m/s^2]
r_earth = 6378;     %[km]
hr2sec  = 60*60;    %[sec/hr]
sec2hr  = 1/hr2sec; %[hr/sec]
deg2rad = pi/180;   %[rad/deg]
rad2deg = 180/pi;   %[deg/rad]
km2m    = 1000;     %[m/km]
m2km    = 1/1000;   %[km/m]
%**********************************************************
nu     = 0;
r      = p/(1+ecc*cos(nu));
P_init = 2*pi*sqrt(r^3/mu);
w_init = 2*pi/P_init;
kk     = 1;
pp     = 1;

%This saves the orbit in the Geocentric Equitorial Frame for one
rotation
%for the initial and final orbit.  This is used to determine the
%necessary thrust vector.
for nu = 0:0.1:2*pi+0.1
    [r_ijk v_ijk] = pqw2ijk(p,ecc,inc1,RAAN,arg0per,nu);
    r_ijk1(kk,:)  = r_ijk;
    v_ijk1(kk,:)  = v_ijk;
    [r_ijk v_ijk] = pqw2ijk(p,ecc,inc2,RAAN,arg0per,nu);
    r_ijk2(kk,:)  = r_ijk;
    v_ijk2(kk,:)  = v_ijk;
    if nu <= pi+0.1
        r_ijk2a(kk,:)  = r_ijk;
        v_ijk2a(kk,:)  = v_ijk;
```

```
    else
        r_ijk2b(pp,:)  = r_ijk;
        v_ijk2b(pp,:)  = v_ijk;
        pp = pp + 1;
    end
    kk = kk+1;
end
for nu = 0:0.1:pi+0.1
    [r_ijk v_ijk] = pqw2ijk(p,ecc,inc2,RAAN,arg0per,nu);
    r_ijk2a(kk,:)  = r_ijk;
    v_ijk2a(kk,:)  = v_ijk;
    kk             = kk+1;
end

h2   = cross(r_ijk2(1,:),v_ijk2(1,:));
inc2 = abs(pi- acos(h2(3)/mag(h2)));
h1   = cross(r_ijk1(1,:),v_ijk1(1,:));
inc1 = abs(pi - acos(h1(3)/mag(h1)));

%*********************************************************************
%**********Thrust Scheme 1: 2 - 15 min burns at 0 and pi***********
%*********************************************************************
%*********************************************************************
mass         = m_init;          %[kg]
t_step       = 15/60*hr2sec;    %[sec]
t_max        = t_const;         %[sec]
kk           = 1;               %[--]
jj           = 1;               %[--]
time         = 0;               %[sec]
v_ijk3(1,:) = v_ijk1(1,:);      %[km/s]
r_ijk3(1,:) = r_ijk1(1,:);      %[km/s]
dv_track     = 0;               %[km/s]
inc          = 0;               %[rad]
inc_tot      = 0;               %[rad]
mp_tot       = 0;               %[kg]

%Calulate initial thrust vector between initial & final orbit
v_ijk_diff = [v_ijk1(1,1) - v_ijk2(1,1),...
              v_ijk1(1,2) - v_ijk2(1,2),...
              v_ijk1(1,3) - v_ijk2(1,3)];

while time < t_max && inc < inc2 && mass > m_init-m_prop

    dv                = Thrust_ep*t_step/mass*m2km;          %[km/s]
    dv_track          = dv_track + dv;                       %[km/s]
    mp                = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
    mp_tot            = mp_tot + mp;                         %[kg]
    mass              = mass - mp;                           %[kg]

    %Calulate updated thrust vector between transfer & final orbit
    [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2a,-dv,t_step);

    %Saves location that thrust occurs at for plot********************
    if time == 0;
        ll = 1;
```

```
    r_ijk3_thrust(ll,:) = r_ijk3;
end
%*****************************************************************
%Calculate updated inclination between transfer & final orbit
h3  = cross(r_ijk3,v_ijk3);
inc = abs(pi - acos(h3(3)/mag(h3)));

%Transfer from Geocentric Equatorial to Perifocal
r_pqw3       = r_ijk3*R313(arg0per,pi-inc,RAAN);
v_pqw3       = v_ijk3*R313(arg0per,pi-inc,RAAN);

for nu = 0:0.1:pi+0.1
    r_pqw(jj,:) = [mag(r_pqw3)*cos(nu) mag(r_pqw3)*sin(nu) 0];
    v_pqw(jj,:) = sqrt(mu/mag(r_pqw3))*[-sin(nu) (ecc+cos(nu)) 0];
    jj = jj + 1;
end

%Saves location that thrust occurs at for plot********************
time    = time + 0.5*P_init;  %Update time
if time == 0.5*P_init;
    ll = ll + 1;
    r_ijk3 = r_pqw(jj-2,:)*R313(RAAN,pi-inc,arg0per);
    r_ijk3_thrust(ll,:) = r_ijk3;
end
%*****************************************************************
%Transfer from Perifocal to Geocentric Equatorial
r_ijk3 = r_pqw*R313(RAAN,pi-inc,arg0per);
v_ijk3 = v_pqw*R313(RAAN,pi-inc,arg0per);

%Second thrust at nu = pi*****************************************
dv               = Thrust_ep*t_step/mass*m2km;         %[km/s]
dv_track         = dv_track + dv;                      %[km/s]
mp               = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
mp_tot           = mp_tot + mp;                        %[kg]
mass             = mass - mp;                          %[kg]

[r_ijk3,v_ijk3]  = updaterv(r_ijk3,v_ijk3,v_ijk2b,-dv,t_step);

%Calculate updated inclination between transfer & final orbit
h3             = cross(r_ijk3,v_ijk3);
inc            = abs(pi- acos(h3(3)/mag(h3)));   %[rad]
inc_tot        = inc_tot + inc;                  %[rad]
inc_track(kk) = inc;                             %[rad]

%Transfer from Geocentric Equatorial to Perifocal
r_pqw3       = r_ijk3*R313(arg0per,pi-inc,RAAN);   %[km]
v_pqw3       = v_ijk3*R313(arg0per,pi-inc,RAAN);   %[km/s]

%Go through remainder of orbit after second thrust
for nu = (0.1+pi):0.1:2*pi
    r_pqw(jj,:) = [mag(r_pqw3)*cos(nu) mag(r_pqw3)*sin(nu) 0];
    v_pqw(jj,:) = sqrt(mu/mag(r_pqw3))*[-sin(nu) (ecc+cos(nu)) 0];
    jj          = jj + 1;
end
r_ijk3 = r_pqw*R313(RAAN,pi-inc,arg0per);
```

```
    v_ijk3 = v_pqw*R313(RAAN,pi-inc,arg0per);
    jj     = 1;
    time   = time + 0.5*P_init;  %[sec] Update time

    Th_prof1 = [inc, time, mp_tot,dv_track];
    r_ijk3_tp1 = r_ijk3;
    v_ijk3_tp1 = v_ijk3;
end

figure(1)
plot3(r_ijk1(:,1),r_ijk1(:,2),r_ijk1(:,3),'xg',r_ijk3(:,1),r_ijk3(:,2),
r_ijk3(:,3),r_ijk2(:,1),r_ijk2(:,2),r_ijk2(:,3),r_ijk3_thrust(:,1),r_ij
k3_thrust(:,2),r_ijk3_thrust(:,3),'s')
title('Final Orbit for 2 - 15 min burns at nu = 0 and pi')
legend('Initial Orbit','Transfer Orbit','Final Orbit','Thrust
Location')

%*******************************************************************
%**************Thrust Scheme 2: 1 - 15 min burn at 0****************
%*******************************************************************
%*********************Re-initialize Variables***********************
mass        = m_init;         %[kg]
t_step      = 18/60*hr2sec;   %[sec]
t_max       = t_const;        %[sec]
jj          = 1;              %[--]
time        = 0;              %[sec]
v_ijk3(1,:) = v_ijk1(1,:);    %[km/s]
r_ijk3(1,:) = r_ijk1(1,:);    %[km/s]
dv_track    = 0;              %[km/s]
inc         = 0;              %[rad]
mp_tot      = 0;              %[kg]
%*******************************************************************

while time < t_max && inc < inc2 && mass > m_init-m_prop
    dv               = Thrust_ep*t_step/mass*m2km;         %[km/s]
    dv_track         = dv_track + dv;                      %[km/s]
    mp               = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
    mp_tot           = mp_tot + mp;                        %[kg]
    mass             = mass - mp;                          %[kg]

    %Calulate updated thrust vector between transfer & final orbit
    [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2a,-dv,t_step);

    %Calculate updated inclination between transfer & final orbit
    h3  = cross(r_ijk3,v_ijk3);
    inc = abs(pi - acos(h3(3)/mag(h3)));

    %Transfer from Geocentric Equatorial to Perifocal
    r_pqw3       = r_ijk3*R313(arg0per,pi-inc,RAAN);    %[km]
    v_pqw3       = v_ijk3*R313(arg0per,pi-inc,RAAN);    %[km/s]

    %Calculate orbit from 0 to 2pi after thrust
    for nu = 0:0.1:2*pi+0.1
        r_pqw(jj,:) = [mag(r_pqw3)*cos(nu) r*sin(nu) 0];
        v_pqw(jj,:) = sqrt(mu/p)*[-sin(nu) (ecc+cos(nu)) 0];
```

```
        jj          = jj + 1;
    end
    jj     = 1;
    r_ijk3 = r_pqw*R313(RAAN,pi-inc,arg0per);    %[km]
    v_ijk3 = v_pqw*R313(RAAN,pi-inc,arg0per);    %[km/s]
    time   = time + P_init;                      %[sec]Update time

    Th_prof2      = [inc, time, mp_tot,dv_track];
    r_ijk3_tp2    = r_ijk3;
    v_ijk3_tp2    = v_ijk3;
    r_ijk3_thrust = r_ijk3(1,:);  %Save Thrust location for plot
end

figure(2)
plot3(r_ijk1(:,1),r_ijk1(:,2),r_ijk1(:,3),'xg',r_ijk3(:,1),r_ijk3(:,2),
r_ijk3(:,3),r_ijk2(:,1),r_ijk2(:,2),r_ijk2(:,3),r_ijk3_thrust(:,1),r_ij
k3_thrust(:,2),r_ijk3_thrust(:,3),'s')
title('Final Orbit for 1 - 15 min burn at nu = 0')
legend('Initial Orbit','Transfer Orbit','Final Orbit','Thrust
Location')

%*******************************************************************
%**********Thrust Scheme 3: 2 - 15 min burns & Chemical***********
%*******************************************************************
%*********************Re-initialize Variables**********************
mass        = m_init;         %[kg]
t_step      = 15/60*hr2sec;   %[sec]
t_max       = t_const;        %[sec]
kk          = 1;              %[--]
jj          = 1;              %[--]
time        = 0;              %[sec]
v_ijk3(1,:) = v_ijk1(1,:);    %[km/s]
r_ijk3(1,:) = r_ijk1(1,:);    %[km/s]
dv_track    = 0;              %[km/s]
inc         = 0;              %[rad]
inc_tot     = 0;              %[rad]
mp_tot      = 0;              %[kg]
%*******************************************************************
while time < t_max && inc < inc2 && mass > m_init-m_prop

    dv               = Thrust_ep*t_step/mass*m2km;         %[km/s]
    dv_track         = dv_track + dv;                      %[km/s]
    mp               = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
    mp_tot           = mp_tot + mp;                        %[kg]
    mass             = mass - mp;                          %[kg]

    %Calulate updated thrust vector between transfer & final orbit
    [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2a,-dv,t_step);

    %Saves location that thrust occurs at for plot*********************
    if time == 0;
        ll = 1;
        r_ijk3_thrust(ll,:) = r_ijk3;
    end
    %*****************************************************************
```

```
%Calculate updated inclination between transfer & final orbit
h3  = cross(r_ijk3,v_ijk3);
inc = abs(pi - acos(h3(3)/mag(h3)));

%Transfer from Geocentric Equatorial to Perifocal
r_pqw3       = r_ijk3*R313(arg0per,pi-inc,RAAN);
v_pqw3       = v_ijk3*R313(arg0per,pi-inc,RAAN);

for nu = 0:0.1:pi+0.1
    r_pqw(jj,:) = [mag(r_pqw3)*cos(nu) mag(r_pqw3)*sin(nu) 0];
    v_pqw(jj,:) = sqrt(mu/mag(r_pqw3))*[-sin(nu) (ecc+cos(nu)) 0];
    jj = jj + 1;
end

%Saves location that thrust occurs at for plot*********************
time    = time + 0.5*P_init;  %Update time
if time == 0.5*P_init;
    ll = ll + 1;
    r_ijk3 = r_pqw(jj-2,:)*R313(RAAN,pi-inc,arg0per);
    r_ijk3_thrust(ll,:) = r_ijk3;
end
%*****************************************************************
%Transfer from Perifocal to Geocentric Equatorial
r_ijk3 = r_pqw*R313(RAAN,pi-inc,arg0per);
v_ijk3 = v_pqw*R313(RAAN,pi-inc,arg0per);

%Second thrust at nu = pi*****************************************
dv                = Thrust_ep*t_step/mass*m2km;         %[km/s]
dv_track          = dv_track + dv;                      %[km/s]
mp                = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
mp_tot            = mp_tot + mp;                        %[kg]
mass              = mass - mp;                          %[kg]

[r_ijk3,v_ijk3]  = updaterv(r_ijk3,v_ijk3,v_ijk2b,-dv,t_step);

%Calculate updated inclination between transfer & final orbit
h3             = cross(r_ijk3,v_ijk3);
inc            = abs(pi- acos(h3(3)/mag(h3)));   %[rad]
inc_tot        = inc_tot + inc;                  %[rad]
inc_track(kk) = inc;                             %[rad]

%Transfer from Geocentric Equatorial to Perifocal
r_pqw3       = r_ijk3*R313(arg0per,pi-inc,RAAN);   %[km]
v_pqw3       = v_ijk3*R313(arg0per,pi-inc,RAAN);   %[km/s]

%Go through remainder of orbit after second thrust
for nu = (0.1+pi):0.1:2*pi
    r_pqw(jj,:) = [mag(r_pqw3)*cos(nu) mag(r_pqw3)*sin(nu) 0];
    v_pqw(jj,:) = sqrt(mu/mag(r_pqw3))*[-sin(nu) (ecc+cos(nu)) 0];
    jj          = jj + 1;
end
r_ijk3 = r_pqw*R313(RAAN,pi-inc,arg0per);
v_ijk3 = v_pqw*R313(RAAN,pi-inc,arg0per);
jj     = 1;
time   = time + 0.5*P_init;  %[sec] Update time
```

```
    %Chemical will complete maneuver it time limit is reached
    if time > (t_max - 3*hr2sec)
        dv = 2*mag(v_ijk2)*sin((inc2 - inc)/2);
        mp = mass*(1-exp(-dv*km2m/Isp_chem/g0));  %[kg]
        inc = inc2;
        mp_tot   = mp_tot + mp;
        dv_track = dv_track + dv;
    end
    r_ijk3     = r_ijk2;
    Th_prof3   = [inc, time, mp_tot,dv_track];
end

figure(3)
plot3(r_ijk1(:,1),r_ijk1(:,2),r_ijk1(:,3),'xg',r_ijk3(:,1),r_ijk3(:,2),
r_ijk3(:,3),r_ijk2(:,1),r_ijk2(:,2),r_ijk2(:,3),r_ijk3_thrust(:,1),r_ij
k3_thrust(:,2),r_ijk3_thrust(:,3),'s')
title('Final Orbit for 2 - 15 min burn at nu = 0 and pi then Chemical')
legend('Initial Orbit','Transfer Orbit','Final Orbit','Thrust
Location')

%*********************************************************************
%***************Thrust Scheme 4: 15 - 1 min burns*****************
%*********************************************************************
mass        = m_init;          %[kg]
t_step      = 15/60*hr2sec;    %[sec]
t_step_min  = 1*60;            %[sec]
t_max       = t_const;         %[sec]
jj          = 1;               %[--]
time        = 0;               %[sec]
v_ijk3(1,:) = v_ijk1(1,:);     %[km/s]
r_ijk3(1,:) = r_ijk1(1,:);     %[km/s]
dv_track    = 0;               %[km/s]
inc         = 0;               %[rad]
mp_tot      = 0;               %[kg]
%************************************************************
%Calulate initial thrust vector between initial & final orbit
v_ijk_diff = [v_ijk1(1,1) - v_ijk2(1,1),...
              v_ijk1(1,2) - v_ijk2(1,2),...
              v_ijk1(1,3) - v_ijk2(1,3)];

while time < t_max && inc < inc2 && mass > m_init-m_prop

    for time2 = 0:t_step_min:t_step
        dv         = Thrust_ep*t_step_min/mass*m2km;     %[km/s]
        dv_track   = dv_track + dv;                      %[km/s]
        mp         = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
        mp_tot     = mp_tot + mp;                        %[kg]
        mass       = mass - mp;                          %[kg]

        %Calulate updated thrust vector between transfer & final orbit
        if time2 > 0
            [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2f,-
dv,time2);
        else
```

```
            [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2,-dv,time2);
        end

        %Calculate updated inclination between transfer & final orbit
        h3  = cross(r_ijk3,v_ijk3);                         %[km^2/s]
        inc = abs(pi - acos(h3(3)/mag(h3)));                %[rad]

        %Transfer from Geocentric Equatorial to Perifocal
        r_pqw3a      = r_ijk3*R313(arg0per,pi-inc,RAAN);    %[km]
        v_pqw3a      = v_ijk3*R313(arg0per,pi-inc,RAAN);    %[km/s]
        nua          = w_init*time2;                        %[rad]
        r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nua) mag(r_pqw3a)*sin(nua) 0];
        v_pqw3(jj,:) = sqrt(mu/mag(r_pqw3a))*[-sin(nua) (ecc+cos(nua))
0];
        r_ijk3       = r_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per);  %[km]
        v_ijk3       = v_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per);  %[km/s]

        r_ijk3_thrust(jj,:) = r_ijk3;

%Calc where final orbit position is to get correct thrust vector
        r_pqw2f = [mag(r_ijk2(1,:))*cos(nua) mag(r_ijk2(1,:))*sin(nua)
0];
        v_pqw2f = sqrt(mu/p)*[-sin(nua) (ecc+cos(nua)) 0];
        r_ijk2f = r_pqw2f*R313(RAAN,pi-inc,arg0per);
        v_ijk2f = v_pqw2f*R313(RAAN,pi-inc,arg0per);

        if time2 == t_step
            r_ijk3 = r_pqw3*R313(RAAN,pi-inc,arg0per);
            v_ijk3 = v_pqw3*R313(RAAN,pi-inc,arg0per);
        end
        jj = jj + 1;
    end
    jj = jj - 1;

    %Calculate orbit after thrust
    for nu = nua:0.1:(2*pi+nua)
        r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nu) mag(r_pqw3a)*sin(nu) 0];
        v_pqw3(jj,:) = sqrt(mu/p)*[-sin(nu) (ecc+cos(nu)) 0];
        jj = jj + 1;
    end
    r_ijk3 = r_pqw3*R313(RAAN,pi-inc,arg0per);
    v_ijk3 = v_pqw3*R313(RAAN,pi-inc,arg0per);
    jj   = jj - 1;
    jj   = 1;
    time = time + P_init;  %Update time, each time through is one orbit

    Th_prof4     = [inc, time, mp_tot,dv_track];
    r_ijk4_tp4   = r_ijk3;
    v_ijk4_tp4   = v_ijk3;
end

figure(4)
plot3(r_ijk1(:,1),r_ijk1(:,2),r_ijk1(:,3),'xg',r_ijk3(:,1),r_ijk3(:,2),
r_ijk3(:,3),r_ijk2(:,1),r_ijk2(:,2),r_ijk2(:,3),r_ijk3_thrust(:,1),r_ij
k3_thrust(:,2),r_ijk3_thrust(:,3),'s')
```

```
title('Final Orbit for 15 - 1 min burns beginning at nu = 0')
legend('Initial Orbit','Transfer Orbit','Final Orbit','Thrust
Location')

%*******************************************************************
%*************Thrust Scheme 5: Continuous 1 min burns**************
%*******************************************************************
%***********************************************************
mass         = m_init;         %[kg]
t_step       = (P_init);       %[sec]
t_step_min   = t_step/60;      %[sec]
t_max        = t_const;        %[sec]
kk           = 1;              %[--]
jj           = 1;              %[--]
yy           = 1;              %[--]
time         = 0;              %[sec]
v_ijk3(1,:) = v_ijk1(1,:);     %[km/s]
r_ijk3(1,:) = r_ijk1(1,:);     %[km/s]
dv_track     = 0;              %[km/s]
inc          = 0;              %[rad]
inc_tot      = 0;              %[rad]
mp_tot       = 0;              %[kg]

%Calulate initial thrust vector between initial & final orbit
v_ijk_diff = [v_ijk1(1,1) - v_ijk2(1,1),...
              v_ijk1(1,2) - v_ijk2(1,2),...
              v_ijk1(1,3) - v_ijk2(1,3)];

while time < t_max && inc < inc2 && mass > m_init-m_prop

    for time2 = 0:t_step_min:t_step
        dv        = Thrust_ep*t_step_min/mass*m2km;     %[km/s]
        dv_track  = dv_track + dv;                      %[km/s]
        mp        = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
        mp_tot    = mp_tot + mp;                        %[kg]
        mass      = mass - mp;                          %[kg]

        %Calulate updated thrust vector between transfer & final orbit
        if time2 == 0
            [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2,-dv,time2);
        elseif time2 > P_init/2
            [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2f,dv,time2);
        elseif time2 < P_init/2
            [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2f,-
dv,time2);
        end

        %Calculate updated inclination between transfer & final orbit
        h3  = cross(r_ijk3,v_ijk3);                       %[km^2/s]
        inc = abs(pi - acos(h3(3)/mag(h3)));              %[rad]

        %Transfer from Geocentric Equatorial to Perifocal
        r_pqw3a      = r_ijk3*R313(arg0per,pi-inc,RAAN);  %[km]
        v_pqw3a      = v_ijk3*R313(arg0per,pi-inc,RAAN);  %[km/s]
        nua          = w_init*time2;                      %[rad]
```

```
        r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nua) mag(r_pqw3a)*sin(nua) 0];
        v_pqw3(jj,:) = sqrt(mu/mag(r_pqw3a))*[-sin(nua) (ecc+cos(nua))
0];
        r_ijk3       = r_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per);  %[km]
        v_ijk3       = v_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per);  %[km/s]

%Calc where final orbit position is to get correct thrust vector
        r_pqw2f = [mag(r_ijk2(1,:))*cos(nua) mag(r_ijk2(1,:))*sin(nua)
0];
        v_pqw2f = sqrt(mu/p)*[-sin(nua) (ecc+cos(nua)) 0];     %[km/s]
        r_ijk2f = r_pqw2f*R313(RAAN,pi-inc,arg0per);           %[km]
        v_ijk2f = v_pqw2f*R313(RAAN,pi-inc,arg0per);           %[km/s]

        if time2 == t_step
            r_ijk3 = r_pqw3*R313(RAAN,pi-inc,arg0per);
            v_ijk3 = v_pqw3*R313(RAAN,pi-inc,arg0per);
        end
        jj   = jj + 1;
    end

    for nu = nua:0.1:(2*pi) %2*pi+nua
        r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nu) mag(r_pqw3a)*sin(nu) 0];
        v_pqw3(jj,:) = sqrt(mu/p)*[-sin(nu) (ecc+cos(nu)) 0];
        jj = jj + 1;
    end
    r_ijk3        = r_pqw3*R313(RAAN,pi-inc,arg0per);    %[km]
    v_ijk3        = v_pqw3*R313(RAAN,pi-inc,arg0per);    %[km/s]
    r_ijk3_thrust = r_ijk3;
    jj            = jj - 1;
    if time == 0
        yy = 1:1:jj;
        yy_old = yy(end);
    else
        yy = yy_old+1:1:jj+yy_old;
        yy_old = yy(end);
    end
    jj   = 1;
    for count = yy(1):1:yy(end)
        r_ijk3_track(count,:) = r_ijk3(jj,:);
        jj = jj +1;
    end
    jj           = 1;
    time         = time + P_init;  %[sec] Update time
    Th_prof5     = [inc, time, mp_tot,dv_track];
    r_ijk4_tp5   = r_ijk3;
    v_ijk4_tp5   = v_ijk3;
end

figure(5)
plot3(r_ijk1(:,1),r_ijk1(:,2),r_ijk1(:,3),'xg',r_ijk3(:,1),r_ijk3(:,2),
r_ijk3(:,3),r_ijk2(:,1),r_ijk2(:,2),r_ijk2(:,3),r_ijk3_thrust(:,1),r_ij
k3_thrust(:,2),r_ijk3_thrust(:,3),'s')
title('Final Orbit for 15 - 1 min burns beginning at nu = 0')
legend('Initial Orbit','Transfer Orbit','Final Orbit','Thrust
Location')
```

```
%*******************************************************************
%***************Thrust Scheme 6: 15 - 1 min burns*******************
%*************Centered around crossover points*********************
%*******************************************************************
mass         = m_init;         %[kg]
t_step       = 16/60*hr2sec;   %[sec]
t_step_min   = 1*60;           %[sec]
t_max        = t_const;        %[sec]
jj           = 1;              %[--]
mm           = 0;              %[--]
time         = 0;              %[sec]
v_ijk3(1,:) = v_ijk1(1,:);    %[km/s]
r_ijk3(1,:) = r_ijk1(1,:);    %[km/s]
dv_track     = 0;              %[km/s]
inc          = 0;              %[rad]
mp_tot       = 0;              %[kg]
%***********************************************************
%Calulate initial thrust vector between initial & final orbit
v_ijk_diff = [v_ijk1(1,1) - v_ijk2(1,1),...
              v_ijk1(1,2) - v_ijk2(1,2),...
              v_ijk1(1,3) - v_ijk2(1,3)];

while time < t_max && inc < inc2 && mass > m_init-m_prop

    for time2 = 0:t_step_min:(t_step/2)
        dv         = Thrust_ep*t_step_min/mass*m2km;     %[km/s]
        dv_track   = dv_track + dv;                      %[km/s]
        mp         = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
        mp_tot     = mp_tot + mp;                        %[kg]
        mass       = mass - mp;                          %[kg]

        %Calulate updated thrust vector between transfer & final orbit
        if time2 > 0
            [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2f,-
dv,time2);
        else
            [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2,-dv,time2);
        end

        %Calculate updated inclination between transfer & final orbit
        h3  = cross(r_ijk3,v_ijk3);                      %[km^2/s]
        inc = abs(pi - acos(h3(3)/mag(h3)));             %[rad]

        %Transfer from Geocentric Equatorial to Perifocal
        r_pqw3a      = r_ijk3*R313(arg0per,pi-inc,RAAN);     %[km]
        v_pqw3a      = v_ijk3*R313(arg0per,pi-inc,RAAN);     %[km/s]
        nua          = w_init*time2;                         %[rad]
        r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nua) mag(r_pqw3a)*sin(nua) 0];
        v_pqw3(jj,:) = sqrt(mu/mag(r_pqw3a))*[-sin(nua) (ecc+cos(nua))
0];
        r_ijk3       = r_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per);  %[km]
        v_ijk3       = v_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per);  %[km/s]

        if time == 0
```

```
            mm = mm + 1;
            r_ijk3_thrust6(mm,:) = r_ijk3;
        end

        %Calc where final orbit position is to get correct thrust
vector
        r_pqw2f = [mag(r_ijk2(1,:))*cos(nua) mag(r_ijk2(1,:))*sin(nua)
0];
        v_pqw2f = sqrt(mu/p)*[-sin(nua) (ecc+cos(nua)) 0];
        r_ijk2f = r_pqw2f*R313(RAAN,pi-inc,arg0per);
        v_ijk2f = v_pqw2f*R313(RAAN,pi-inc,arg0per);

        if time2 == t_step
            r_ijk3 = r_pqw3*R313(RAAN,pi-inc,arg0per);
            v_ijk3 = v_pqw3*R313(RAAN,pi-inc,arg0per);
        end
        jj = jj + 1;
    end
    jj = jj - 1;

    %Calculate orbit after thrust
    for nu = nua:0.1:(2*pi-nua)
        r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nu) mag(r_pqw3a)*sin(nu) 0];
        v_pqw3(jj,:) = sqrt(mu/p)*[-sin(nu) (ecc+cos(nu)) 0];
        jj = jj + 1;
    end
    nua    = nu;
    time2 = nua/w_init;
    r_ijk3 = r_pqw3*R313(RAAN,pi-inc,arg0per);
    v_ijk3 = v_pqw3*R313(RAAN,pi-inc,arg0per);

    for time2 = (time2):t_step_min:(time2+t_step/2)
        dv        = Thrust_ep*t_step_min/mass*m2km;      %[km/s]
        dv_track  = dv_track + dv;                       %[km/s]
        mp        = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
        mp_tot    = mp_tot + mp;                         %[kg]
        mass      = mass - mp;                           %[kg]

        %Calulate updated thrust vector between transfer & final orbit
        [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2f,dv,time2);

        %Calculate updated inclination between transfer & final orbit
        h3  = cross(r_ijk3,v_ijk3);                     %[km^2/s]
        inc = abs(pi - acos(h3(3)/mag(h3)));            %[rad]

        %Transfer from Geocentric Equatorial to Perifocal
        r_pqw3a      = r_ijk3*R313(arg0per,pi-inc,RAAN);    %[km]
        v_pqw3a      = v_ijk3*R313(arg0per,pi-inc,RAAN);    %[km/s]
        nua          = w_init*time2;                        %[rad]
        r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nua) mag(r_pqw3a)*sin(nua) 0];
        v_pqw3(jj,:) = sqrt(mu/mag(r_pqw3a))*[-sin(nua) (ecc+cos(nua))
0];
        r_ijk3       = r_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per);    %[km]
        v_ijk3       = v_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per);
%[km/s]
```

```
        if time == 0
            mm = mm + 1;
            r_ijk3_thrust6(mm,:) = r_ijk3;
        end

        %Calc where final orbit position is to get correct thrust
vector
        r_pqw2f = [mag(r_ijk2(1,:))*cos(nua) mag(r_ijk2(1,:))*sin(nua)
0];
        v_pqw2f = sqrt(mu/p)*[-sin(nua) (ecc+cos(nua)) 0];
        r_ijk2f = r_pqw2f*R313(RAAN,pi-inc,arg0per);
        v_ijk2f = v_pqw2f*R313(RAAN,pi-inc,arg0per);

        if time2 == t_step
            r_ijk3 = r_pqw3*R313(RAAN,pi-inc,arg0per);
            v_ijk3 = v_pqw3*R313(RAAN,pi-inc,arg0per);
        end
        jj = jj + 1;
    end
    r_ijk3   = r_pqw3*R313(RAAN,pi-inc,arg0per);    %[km]
    v_ijk3   = v_pqw3*R313(RAAN,pi-inc,arg0per);    %[km]
    jj       = 1;                                   %[--]
    time     = time + P_init;  %[sec] Update time
    Th_prof6 = [inc, time, mp_tot,dv_track];
end

figure(6)
plot3(r_ijk1(:,1),r_ijk1(:,2),r_ijk1(:,3),'xg',r_ijk3(:,1),r_ijk3(:,2),
r_ijk3(:,3),r_ijk2(:,1),r_ijk2(:,2),r_ijk2(:,3),r_ijk3_thrust6(:,1),r_i
jk3_thrust6(:,2),r_ijk3_thrust6(:,3),'s')
title('Final Orbit for 15 - 1 min burns centered at nu = 0')
legend('Initial Orbit','Transfer Orbit','Final Orbit','Thrust
Location')

%*********************************************************************
%***************Thrust Scheme 7: 15 - 1 min burns******************
%*************Centered nu = pi/2 and 3pi/4************************
%*********************************************************************
mass        = m_init;          %[kg]
t_step      = 16/60*hr2sec;    %[sec]
t_step_min  = 1*60;            %[sec]
t_max       = t_const;         %[sec]
jj          = 1;               %[--]
mm          = 0;               %[--]
time        = 0;               %[sec]
v_ijk3(1,:) = v_ijk1(1,:);     %[km/s]
r_ijk3(1,:) = r_ijk1(1,:);     %[km/s]
dv_track    = 0;               %[km/s]
inc         = 0;               %[rad]
mp_tot      = 0;               %[kg]
%*************************************************************
%Calulate initial thrust vector between initial & final orbit
v_ijk_diff = [v_ijk1(1,1) - v_ijk2(1,1),...
              v_ijk1(1,2) - v_ijk2(1,2),...
```

```
              v_ijk1(1,3) - v_ijk2(1,3)];

while time < t_max && inc < inc2 && mass > m_init-m_prop

    for time2 = 0:t_step_min:(t_step/2)
        dv         = Thrust_ep*t_step_min/mass*m2km;      %[km/s]
        dv_track   = dv_track + dv;                       %[km/s]
        mp         = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
        mp_tot     = mp_tot + mp;                         %[kg]
        mass       = mass - mp;                           %[kg]

        %Calulate updated thrust vector between transfer & final orbit
        if time2 > 0
            [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2f,-
dv,time2);
        else
            [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2,-dv,time2);
        end

        %Calculate updated inclination between transfer & final orbit
        h3  = cross(r_ijk3,v_ijk3);                      %[km^2/s]
        inc = abs(pi - acos(h3(3)/mag(h3)));             %[rad]

        %Transfer from Geocentric Equatorial to Perifocal
        r_pqw3a      = r_ijk3*R313(arg0per,pi-inc,RAAN);    %[km]
        v_pqw3a      = v_ijk3*R313(arg0per,pi-inc,RAAN);    %[km/s]
        nua          = w_init*time2;                        %[rad]
        r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nua) mag(r_pqw3a)*sin(nua) 0];
        v_pqw3(jj,:) = sqrt(mu/mag(r_pqw3a))*[-sin(nua) (ecc+cos(nua))
0];
        r_ijk3       = r_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per);    %[km]
        v_ijk3       = v_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per);
%[km/s]

        if time == 0
            mm = mm + 1;
            r_ijk3_thrust6(mm,:) = r_ijk3;
        end

        %Calc where final orbit position is to get correct thrust
vector
        r_pqw2f = [mag(r_ijk2(1,:))*cos(nua) mag(r_ijk2(1,:))*sin(nua)
0];
        v_pqw2f = sqrt(mu/p)*[-sin(nua) (ecc+cos(nua)) 0];
        r_ijk2f = r_pqw2f*R313(RAAN,pi-inc,arg0per);
        v_ijk2f = v_pqw2f*R313(RAAN,pi-inc,arg0per);

        if time2 == t_step
            r_ijk3 = r_pqw3*R313(RAAN,pi-inc,arg0per);
            v_ijk3 = v_pqw3*R313(RAAN,pi-inc,arg0per);
        end
        jj = jj + 1;
    end
    jj = jj - 1;
```

```
    %Calculate orbit after thrust
    for nu = nua:0.1:(2*pi-nua)
        r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nu) mag(r_pqw3a)*sin(nu) 0];
        v_pqw3(jj,:) = sqrt(mu/p)*[-sin(nu) (ecc+cos(nu)) 0];
        jj = jj + 1;
    end
    nua    = nu;
    time2 = nua/w_init;
    r_ijk3 = r_pqw3*R313(RAAN,pi-inc,arg0per);
    v_ijk3 = v_pqw3*R313(RAAN,pi-inc,arg0per);

    for time2 = (time2):t_step_min:(time2+t_step/2)
        dv         = Thrust_ep*t_step_min/mass*m2km;     %[km/s]
        dv_track   = dv_track + dv;                      %[km/s]
        mp         = mass*(1-exp(-dv*km2m/(Isp_ep*g0))); %[kg]
        mp_tot     = mp_tot + mp;                        %[kg]
        mass       = mass - mp;                          %[kg]

        %Calulate updated thrust vector between transfer & final orbit
        [r_ijk3,v_ijk3] = updaterv(r_ijk3,v_ijk3,v_ijk2f,dv,time2);

        %Calculate updated inclination between transfer & final orbit
        h3  = cross(r_ijk3,v_ijk3);                     %[km^2/s]
        inc = abs(pi - acos(h3(3)/mag(h3)));            %[rad]

        %Transfer from Geocentric Equatorial to Perifocal
        r_pqw3a      = r_ijk3*R313(arg0per,pi-inc,RAAN);    %[km]
        v_pqw3a      = v_ijk3*R313(arg0per,pi-inc,RAAN);    %[km/s]
        nua          = w_init*time2;                        %[rad]
        r_pqw3(jj,:) = [mag(r_pqw3a)*cos(nua) mag(r_pqw3a)*sin(nua) 0];
        v_pqw3(jj,:) = sqrt(mu/mag(r_pqw3a))*[-sin(nua) (ecc+cos(nua))
0];
        r_ijk3       = r_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per); %[km]
        v_ijk3       = v_pqw3(jj,:)*R313(RAAN,pi-inc,arg0per); %[km/s]

        if time == 0
            mm = mm + 1;
            r_ijk3_thrust6(mm,:) = r_ijk3;
        end

%Calc where final orbit position is to get correct thrust vector
        r_pqw2f = [mag(r_ijk2(1,:))*cos(nua) mag(r_ijk2(1,:))*sin(nua)
0];
        v_pqw2f = sqrt(mu/p)*[-sin(nua) (ecc+cos(nua)) 0];
        r_ijk2f = r_pqw2f*R313(RAAN,pi-inc,arg0per);
        v_ijk2f = v_pqw2f*R313(RAAN,pi-inc,arg0per);

        if time2 == t_step
            r_ijk3 = r_pqw3*R313(RAAN,pi-inc,arg0per);
            v_ijk3 = v_pqw3*R313(RAAN,pi-inc,arg0per);
        end
        jj = jj + 1;
    end
```

```
    r_ijk3   = r_pqw3*R313(RAAN,pi-inc,arg0per);    %[km]
    v_ijk3   = v_pqw3*R313(RAAN,pi-inc,arg0per);    %[km]
    jj       = 1;                                   %[--]
    time     = time + P_init;                       %[sec] Update time
    Th_prof7 = [inc, time, mp_tot,dv_track];
end
```